

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

**Кафедра системного програмування і
спеціалізованих комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

_____ В.П. Тарасенко

«___» _____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050102 «Комп'ютерна інженерія»

на тему:

**«Пристрій на базі GPS для вимірювання динаміки автомобіля. Додаток
для Android»**

Виконав :

студент IV курсу, групи KB-52

Пойдюк Віталій В'ячеславович

Керівник:

Консультант з нормоконтролю:

Рецензент:

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

**Кафедра системного програмування і
спеціалізованих комп'ютерних систем**

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) –
6.050102 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В.П. Тарасенко

«___» _____ 2019 р.

ЗАВДАННЯ

на дипломний проект студента

Пойдюка Віталія В'ячеславовича

1. Тема проекту «Пристрій на базі GPS для вимірювання динаміки автомобіля. Додаток для Android», керівник проекту Коляда Костянтин В'ячеславович, асистент кафедри СПСКС, затверджені наказом по університету від 22.05.2019 р. №1330-С
2. Термін подання студентом проекту «___» червня 2019 р.
3. Вихідні дані до проекту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - аналіз існуючих рішень вимірювання динаміки автомобіля та обґрунтування теми дипломного проекту;
 - дослідження і аналіз архітектурних аспектів при розробці пристрою на базі GPS для вимірювання динаміки автомобіля;
 - проектування та організація структури пристрою;
 - опис технологій пристрою;
5. Перелік обов'язкового графічного матеріалу:
 - структурна модель;
 - функціональна схема;
 - схема алгоритму роботи додатку;

— принципова схема.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль			

7. Дата видачі завдання «__» _____ 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	01.04.2019	
2.	Розроблення технічного завдання	04.04.2019	
3.	Розроблення структури пристрою на базі GPS	09.04.2019	
4.	Розроблення функціоналу обробки даних отриманих з GPS модуля	14.04.2019	
5.	Підготовка матеріалів першого розділу дипломного проекту	24.04.2019	
6.	Долучення до пристрою акселерометра та розробка функціоналу для роботи з ним.	30.04.2019	
7.	Підготовка матеріалів другого розділу дипломного проекту	07.05.2019	
8.	Розробка додатку для Android	13.05.2019	
9.	Тестування додатку та апаратного забезпечення	19.05.2019	
10.	Підготовка матеріалів третього розділу дипломного проекту	24.05.2019	
11.	Підготовка графічної частини дипломного проекту	30.05.2019	
12.	Оформлення документації дипломного проекту	04.06.2019	

Студент

В.В. Пойдюк

Керівник проекту

К.В. Коляда

АНОТАЦІЯ

Об'єкт розробки – розробка власного пристрою та програмного забезпечення для організації передачі даних з нього. Створення програмного забезпечення для мобільної платформи Android для обробки отриманих даних з приладу.

Розроблений пристрій дозволяє:

- Отримувати швидкість за допомогою GPS модуля;
- Обчислювати 3-ох осьове прискорення автомобіля акселерометром;
- Вести контроль за нахилом руху автомобіля за допомогою гіроскопу;
- Передавати дані за допомогою Bluetooth або Wi-Fi;
- Програмно відключати GPS, акселерометр та гіроскоп.

В пристрою передбачений захист від повного розряду батареї та обробка результатів на основі отриманих даних. В процесі розробки була використана мова програмування C++, Java з використанням бібліотеки SoftSerial та середовище розробки Arduino, Android Studio.

В ході виконання дипломного проекту:

- розроблено апаратний засіб на основі GPS модуля;
- проведено аналіз існуючих рішень вимірювання динаміки автомобіля;
- розроблено програмне забезпечення для підтримки роботи АЗ.

Основною ціллю даного проекту є створення недорого порівняно з вже представленими на ринку пристрою для вимірювання швидкості, який зможе легко конкурувати з представниками цієї ніші. І буде легкодоступним на території України.

Ключові слова: GPS, ESP32, C++, JAVA, ДИНАМІКА, ПРИСКОРЕННЯ, АКСЕЛЕРОМЕТР, ГІРОСКОП, ЧАСТОТА ОНОВЛЕННЯ, ARDUINO, ANDROID.

ABSTRACT

Object of project is development own hardware and software for transfer data by itself. Creating own software for mobile platform for using data from device.

Created device allow:

- Get speed using GPS;
- Counting of three axis car acceleration;
- Look for changing in height using gyroscope;
- Send data using Bluetooth or Wi-Fi;
- Programmatically disable GPS, accelerometer with Bluetooth or Wi-Fi

Deep discharge protection and processing results using data. In development were used programming languages C++, Java. SoftSerial library and Arduino IDE, Android Studio were used.

In time of development:

- Created hardware based on GPS module;
- Analyzed of the finished solutions;
- Created software for working with hardware.

The main goal of this project is to create a low-cost, compared with the already presented on the market measuring device, which can easily compete with the representatives of this niche. And will be easily accessible on the territory of Ukraine.

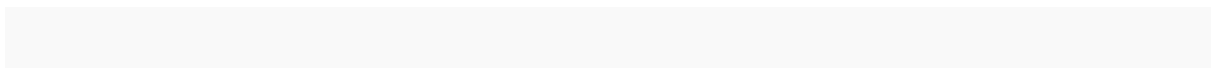
Key words: GPS, ESP32, C++, JAVA, PERFORMANCE METTER, ACCELERATION, ACCELEROMETER, GYROSCOPE, REFRESH RATE, ARDUINO, ANDROID.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			Android			
			Схема пристрою. Схема			
			структурна			
	A4	ІАЛЦ. 045490.006 Д1	Пристрій на базі GPS для	1		
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Схеми алгоритмів			
			бездротової передачі даних			
			Схема структурна			
	A4	ІАЛЦ. 045490.007 Д1	Пристрій на базі GPS для	1		
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Алгоритм роботи додатку			
			для Android. Схема			
			структурна			
	A4	ІАЛЦ. 045490.008 Е3	Пристрій на базі GPS для	1		
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Взаємодія мікроконтролера			
			ESP32 та датчика MPU6050			
			Схема принципова			

					ІАЛЦ. 045490.001 ОА		Арк.
Змін.	Арк.	№ докум.	Підпис	Дата			2

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до програмного продукту, що розробляється	2
5.2. Вимоги до апаратного забезпечення	3
5.3. Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	4



					ІАЛЦ. 045490.002 ТЗ									
Зм	Лист	№ докум.	Підп.	Дата	Технічне завдання					Лім.	Лист	Листів		
Розроб.		Пойдюк												
Перев.		Коляда											1	4
Н. контр.		Клятченко												
Затв.		Тарасенко								НТУУ «КПІ ім. Ігоря Сікорського», ФПМ, КВ-52				

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Пристрій на базі GPS для вимірювання динаміки автомобіля. Додаток для Android».

Галузь застосування: тестування можливостей транспортних засобів.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи ступеня «бакалавр комп'ютерної інженерії», затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення пристрою для вимірювання динаміки автомобіля, а також створення мобільного додатку для операційної системи Android для збору даних з пристрою та їх відображення.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Android;
- можливість вибору проміжних результатів;
- можливість видалення конкретних результатів;
- можливість збереження отриманих даних локально.

					ІАЛЦ.045490.002 ТЗ	Лист 2
Зм	Лист	№ докум.	Підп.	Дата		

5.2. Вимоги до апаратного забезпечення

- Процесор: 32-бітний 2-ядерний Tensilica Xtensa LX6;
- Оперативна пам'ять: 520 Кб;
- Постійний запам'ятовуючий пристрій: 448 Кб;
- GPS з частотою 19 Гц;
- Безпроводний зв'язок: Bluetooth 4.2, BLE, WiFi.

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Android 6.0 +
- Наявність доступу до мережі Internet (GPRS, EDGE, 3G, 4G)

					ІАЛЦ.045490.002 ТЗ	Лист
						3
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту
1.	Вивчення літератури за тематикою проекту	01.04.2019
2.	Розроблення технічного завдання	04.04.2019
3.	Розроблення структури пристрою на базі GPS	09.04.2019
4.	Розроблення функціоналу обробки даних отриманих з GPS модуля	14.04.2019
5.	Підготовка матеріалів першого розділу дипломного проекту	24.04.2019
6.	Долучення до пристрою акселерометра та розробка функціоналу для роботи з ним.	30.04.2019
7.	Підготовка матеріалів другого розділу дипломного проекту	07.05.2019
8.	Розробка додатку для Android	13.05.2019
9.	Тестування додатку та апаратного забезпечення	19.05.2019
10.	Підготовка матеріалів третього розділу дипломного проекту	24.05.2019
11.	Підготовка графічної частини дипломного проекту	30.05.2019
12.	Оформлення документації дипломного проекту	04.06.2019

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
			<u>Документація загальна</u>			
			<u>Новорозроблена</u>			
	A4	ІАЛЦ. 045490.004 ПЗ	Пристрій на базі GPS для	59		
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Пояснювальна записка			
	A4	ІАЛЦ. 045490.005 Д1	Пристрій на базі GPS для	1		
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Схема пристрою. Схема			
			структурна			
	A4	ІАЛЦ. 045490.006 Д1	Пристрій на базі GPS для			
			вимірювання динаміки			
			автомобіля. Додаток для			
			Android			
			Схеми алгоритмів			
			бездротової передачі даних			
			Схема структурна			
	A4	ІАЛЦ. 045490.007 ДЗ	Пристрій на базі GPS для			

					ІАЛЦ. 045490.001 ОА					
Зм	Лист	№ докум.	Підп	Дата	Пристрій на базі GPS для вимірювання динаміки автомобіля. Додаток для Android			Літ.	Лист	Листів
Розроб.		Пойдюк В.В.								
Перев.		Коляда К.В.							1	2
								КПІ ім. Ігоря Сікорського, ФПМ, КВ-52		
Н. контр.		Клятченко Я.М.			Відомість технічного проекту					
Затв.		Тарасенко В.П.								

[illegible]

ЗМІСТ

ЗМІСТ.....	1
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП.....	4
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ, ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМУ.....	5
1.1. Загальний опис проблеми.....	5
1.2. Аналіз існуючих рішень.....	6
1.3. Обґрунтування теми диплому.....	12
2. ОСОБЛИВОСТІ РОЗРОБКИ ПРИСТРОЮ НА БАЗІ GPS ДЛЯ ВИМІРЮВАННЯ ДИНАМІКИ АВТОМОБІЛЯ	13
2.1. Аналіз особливостей розробки.....	13
2.2. Архітектурні аспекти DRT.....	14
2.2.1. Пристрій на базі процесору Atmega 328p.....	14
2.2.2. Пристрій під управлінням Arduino Nano.....	16
2.2.3. Пристрій під управлінням ESP32.....	19
2.3. Проектування та організація структури DRT	21
2.4. Технологія пристрою	24
2.5. Типова реалізація. Приклад розробки	29
3. РЕАЛІЗАЦІЯ ПРОЕКТУ	40
3.1 Вибір схеми моделі	40
3.2 Вибір інструментального програмного забезпечення	43
3.3 Опис архітектури та принцип роботи.....	48
3.4 Опис інтерфейсу.....	53
4. ВИСНОВОКИ.....	58
5. СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	59

					ІАЛЦ.045490.004 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Пристрій на базі GPS для вимірювання динаміки автомобіля. Додаток для Android. Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив	Пойдюк В. В.							
Перев.	Коляда К.В.						1	59
Н. контр.	Клятченко Я.М.					НТУУ "КПІ", ФПМ, КВ-52		
Затвер.	Тарасенко В.П.							

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

ІАЛЦ.045490.005 Д1. Схема пристрою. Схема структурна.

ІАЛЦ.045490.006 Д1. Схеми алгоритмів бездротової передачі даних.
Схема структурна.

ІАЛЦ.045490.007 Д1. Алгоритм роботи додатку. Схема структурна.

ІАЛЦ.045490.008 Е3. Взаємодія мікроконтролера ESP32 та датчика MPU6050. Принципова схема.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

AЗ – апаратне забезпечення

ПЗ – програмне забезпечення

BLE (Bluetooth Low Energy) – блютуз з низьким енергоживленням

GFSK (Gaussian Frequency Shift Keying) – гаусівська частотна модуляція

I2C (Inter-Integrated Circuit) – між-інтегральна схема

PWM (Pulse-Width Modulation) – широтно-імпульсна модуляція

SPI (Serial Peripheral Interface) – послідовний периферійний інтерфейс

USART – універсальний синхронний/асинхронний приймач/передатчик

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

ВСТУП

Стрімкий розвиток мобільних технологій дає змогу розробникам створювати власні програмні та апаратні засоби для вимірювання швидкісних характеристик автомобіля. Ці засоби називають вимірювачами потужності (performance meter). Вимірювач потужності - це апаратний та програмний засіб, що характеризує динамічні можливості автомобіля. Ці пристрої дають змогу визначати час прискорення, проходження дистанцій та перевантаження під час прискорення, кут нахилу траєкторії і так далі.

Метою цих вимірювань є отримання значень, аналізуючи які можна дізнатися розгін автомобіля та гальмівний показник, а також оцінити співвідношення маси до потужності, дорожніх умов, погодних показників та гуми.

Серед усіх пристроїв представлених на ринку найпопулярнішими є Racelogic, Dragy GPS, DragOn, Racebox, Speedlog та інші.

Також для якісного порівняння характеристик автомобіля використовується такі засоби, як акселерометр, гіроскоп та термометр.

Кожний пристрій має своє програмне забезпечення, яке відповідно реалізовано по різному, залежачи від пристрою і тому результати вимірювань навіть на одному автомобілі можуть відрізнятися один від одного.

В даному дипломному проекті описано АЗ, що приймає дані від GPS, акселерометра з гіроскопом та термометром та передає отримані дані на додаток користувача по протоколу Bluetooth 4.0 BLE та ПЗ який відповідно отримує дані від пристрою, оброблює і відображає їх.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Загальний опис проблеми

Проблема що полягає в даному дипломному проекті - розробка дешевого пристрою для вимірювання динаміки автомобіля, який буде легкодоступний в нашій країні.

На даний момент на ринку приладів для моторспорту достатньо просто знайти потрібний апаратний засіб, але дуже складно знайти в нашій країні. Для покупки потрібного пристрою необхідно його замовляти з-за кордону, що тягне за собою додаткові витрати на оплату доставки і розмитнення, а це близько 30% від початкової вартості товару. До того ж на доставку потрібно чекати впродовж 3-5 тижнів. Купуючи засіб вже в Україні продавець вже враховує додаткові витрати на доставку та розмитнення, а доставка до замовника займає набагато менше часу.

Найперші пристрої для вимірювання динаміки автомобіля були досить громіздкими, не мали програми для телефону, що не давало змоги подивитися попередні результати і не мали автономного живлення.

Останні 5 років на ринок було представлено багато нових продуктів від різних виробників, які були вже значно компактнішими, але все ще залишалися дуже не зручними і досить дорогими. Все ще не було додатку для телефону відповідно не було можливості переглянути попередні результати і живлення залишалися не автономним.

Два роки тому китайським розробникам вдалося створити пристрій, який був значно кращим за засоби, які було представлені на ринку до нього. Новий продукт був значно компактнішим, мав вбудовану батарею, що давало змогу пристрою працювати протягом 7-8 годин, також для нього був зроблений додаток для iOS та Android. Роздрібна ціна менша в 4 рази. Але у

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

пристрою залишилися недоліки - практична відсутність в магазинах, тому що вартість розмитнення у відсотковому співвідношенні більша.

Досить важливим фактором все ще залишається сприйняття людьми цих пристроїв, тому що за еталонні значення всі беруть показники пристрою розробленого в далекому 2001 першому році. Не зважаючи уваги на те, що сучасні пристрою мають точніші вимірювання за рахунок частішого оновлення даних та використання сучасних технологій при розробці.

Отже, можна виділити наступні проблеми:

- висока ціна;
- низька конкурентність на ринку
- дороге розмитнення
- довготривала доставка
- не можливість оновлення ПЗ для апаратного засобу

1.2 Аналіз існуючих рішень





Із існуючих пристроїв слід виділити наступні 4 прилади : Racebox, Speedlog, Dragy, DragOn. Подальший аналіз буде базуватися на їх порівнянні.

Також існує багато інших пристроїв. В цілому вони повторюють функціонал DragOn. Майже всі прилади працюють на частоті оновлень 10 Гц і всі мають вбудовану GPS антену.

Слід виділити пристрій на базі ELM327, що являє собою OBD конектор і отримує дані про швидкість з автомобіля. Оскільки дані про швидкість можуть різнитися в залежності від розміру коліс автомобіля та передатних чисел коробки переключення передач. Цей пристрій може допомогти в рішенні наступних задач:

- визначення несправності датчиків двигуна;
- зчитування і перегляд кодів;
- скидання помилок в пам'яті комп'ютера.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

	Racebox	Speedlog	Dragy	DragOn
				
Частота оновлень	10 Hz	10 Hz	10 Hz	10 Hz
Вбудована батарея	Ні	Ні	Так	Ні
Вбудована GPS антена	Так	Так	Так	Так
Накопичувач пам'яті	Внутрішня пам'ять	SD карта	Внутрішня пам'ять	SD карта
Обмін даними через Bluetooth	Ні	Ні	Так	Ні
Додаток для телефону	Ні	Ні	Так	Ні
Розмір	120x80x60 мм	87x57x48 мм	76x25x14 мм	95x33x22 мм
Вага	160 грам	130 грам	105 грам	120 грам
Ціна	110\$	155\$	150\$	200\$
Виміри часу кола	Так	Так	Ні	Ні
Мат. корпусу	АБС пластик	АБС пластик	Полікарбонат	АБС пластик
Дисплей	Так	Так	Ні	Так

Racebox

Технічні характеристики та специфікації

Швидкість:

- Точність: 0,1 км/год
- Одиниці вимірювання: км/год, мл/год
- Частота оновлень: 10 Гц
- Мінімальна швидкість: 0,1 км/год
- Максимальна швидкість: 300 км/год

Прискорення:

- Точність: 1%
- Максимум: 4 G

Вага: 160 грам

Розмір: 120 мм x 80 мм x 60 мм

Можливості пристрою та види замірів :

- розрахункова потужність двигуна;
- зберігання результатів в внутрішній пам'яті пристрою;
- час проходження кола на кільцевій трасі з контролем кращого часу кола;
- контроль схилу дороги;
- запис лога поїздки в форматах CSV, GPX і VBO

Speedlog

Технічні характеристики та специфікації

Швидкість:

- Точність: 0,1 км/год
- Одиниці вимірювання: км/год або мл/год
- Частота оновлень: 10 Гц
- Мінімальна швидкість: 0,1 км/год
- Максимальна швидкість: 300 км/год

Прискорення:

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

- Точність: 1%
- Максимум: 4 G

Вага: 160 грам

Розмір: 87 мм x 57 мм x 48 мм

Можливості пристрою та види замірів :

- є можливість збереження кращих результатів;
- зберігання результатів в внутрішній пам'яті пристрою;
- запис лога поїздки;
- вимірювання часу кола.

Dragy

Технічні характеристики та специфікації

Швидкість:

- Точність: 0,01 км/год
- Одиниці вимірювання: км/год, мл/год
- Частота оновлення: 10 Гц
- Мінімальна швидкість: 0,01 км/год
- Максимальна швидкість: 360 км/год

Прискорення:

- Точність: 1%
- Максимум: 5G

Вага: 105 грам

Розмір: 76 мм x 25 мм x 14 мм

Можливості пристрою та види замірів :

- фіксує кут нахилу під час всього заміру;
- є програма для iOS та Android;
- у додатку iOS є можливість запису відео динаміки;
- дані заїздів зберігаються у форматі метрик і SAE;
- є можливість запису логів;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

- влаштована батарея дає змогу пристрою працювати на протязі 10 годин.

DragOn

Технічні характеристики та специфікації

Швидкість:

- Точність: 0,1 км/год
- Одиниці вимірювання: км/год
- Частота оновлення: 10 Гц
- Мінімальна швидкість: 0,1 км/год
- Максимальна швидкість: 360 км/год

Вага: 120 грам

Розмір: 95 мм x 33 мм x 22 мм

Можливості пристрою та види замірів :

- пристрій може заміряти будь-яких проміжків швидкості від 0 до 300 км/год з кроком в 10 км/год;
- після замірів можна зупинитися і продивитися всі значення;
- можливість запису логу;
- максимальне значення таймеру 99 секунд;

Слід зазначити, що пристрій Dragy може проводити заміри на проміжку 100-200 км/год без повної зупинки автомобіля на відміну від Racebox, Speedlog і DragonOn, які такої можливості не мають. Також потрібно зазначити, що у Dragy відсутній вбудований процесор і тому дані, які надходять на GPS і GLONASS модулі одразу надсилаються через Bluetooth на телефон, де і відбуваються всі обчислення. За рахунок того що обчислення відбуваються на більш потужному телефоні, а не на пристрої є змога отримувати більш точні дані, оскільки затримка в обробці даних значно менша.

До динаміки автомобіля відносять заміри розгону автомобіля від 0 до 100 км/год, а також виконуються тестування гальмівної динаміки. Гальмівна

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

динаміка заміряє час гальмування зі 100 до 0 км/год. Всі прилади, що були описані вище також мають змогу заміряти гальмівну динаміку. Але ці заміри не були описані, тому що ці виміри не є об'єктивними. Ці вимірювання не слід брати до уваги, тому що користувач може заміряти динаміку автомобіля на четверть милі, а потім продовжувати їхати в спокійному режимі, а зупинитися вже на світлофорі. Тому для коректного виміру даного типу динаміки треба використовувати акселерометр, як і для заміру прискорення. За допомогою акселерометра можна точно визначити момент старту, тому що прилад починає відлік часу про розгін і дає можливість побачити яке перевантаження відчуває користувач в початковий момент. За тією ж логікою потрібно заміряти гальмівну динаміку. Автомобіль повинен їхати зі швидкістю 100 км/год за даними пристрою, а не за показниками спідометру, потім водію необхідно натиснути на гальма до повної зупинки машини.

Також треба сказати про програмні забезпечення, які використовують вбудований в телефон GPS модуль. Очевидно, що ці заміри далекі від ідеалу. Операційна система телефону працює з перериваннями і дані про швидкість можуть приходити зі значною затримкою до того ж операційна система може відключати GPS модуль. Тому дані про заїзд будуть не об'єктивними, тому що в результати заміру не враховуються можливі затримки через переривання, що виникають.

Також потрібно відмітити OBD пристрій, що базується на чіпі ELM327. Цей сканер призначений для зчитування інформації з електронного блоку керування автомобіля. За його допомогою можна визначити і усунути деякі несправності двигуна ще на ранній стадії, що пізніше дозволить запобігти серйозні поломки. Цим пристроєм можуть користуватися навіть люди, які не мають професійних знань в даній області. Але не зважаючи навіть на те що пристрій працює без переривань затримка в передачі даних через Bluetooth досить велика, а час засікає телефон, тому таким вимірюванням також не можна довіряти.

1.3 Обґрунтування теми диплому

Головна ціль дипломного проекту - створити конкурентно спроможний пристрій для вимірювання динаміки автомобіля. Проаналізувавши всі доступні продукти на ринку та врахувавши всі їхні переваги та недоліки, були виділені основні апаратні та програмні вимоги для реалізації свого пристрою.

Ключові вимоги до апаратного та програмного засобу:

- висока швидкість оновлення даних з GPS-пристрою;
- зручна програма для телефону(планшету) користувача;
- можливість оновлення програмного забезпечення для апаратного засобу;
- компактний корпус для апаратного засобу;
- довготривала автономна робота пристрою від вбудованої акумуляторної батареї;
- швидкий обмін даними між пристроєм і програмним додатком
- доступна ціна;

Задача полягає в створенні апаратного забезпечення та розробки програмного забезпечення, яке буде взаємодіяти з апаратним засобом. Розробка додатку для мобільної платформи Android.

Створення особистого апаратного засобу полягає в бажанні створити прилад, який складе достойну конкуренцію іншим пристроям представленим на ринку і в першу чергу зможе показати якісно точніші результати ніж конкуренти. Власна платформа дасть змогу розробнику оптимізувати програмне забезпечення для апаратного засобу та цим якісно конкурувати з діючими лідерами даної сфери на ринку засобів для моторспорту.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		12

2. ОСОБЛИВОСТІ РОЗРОБКИ ПРИСТРОЮ НА БАЗІ GPS ДЛЯ ВИМІРЮВАННЯ ДИНАМІКИ АВТОМОБІЛЯ

2.1 Аналіз особливостей даної розробки

Для зручності висловлювань та подальшого комерційного використання дамо назву проекту – DRT (drag racing tool).

В ході розробки даного проекту було використано багато ключових рішень. До таких рішень можна віднести:

- використання плати ESP32 в якій є вбудований Bluetooth/Wi-Fi модуль. Це дає змогу підключити GPS до найпотужніших входів на платі, що означає обробку даних на основному ядрі, яке є потужнішим за допоміжне. Для обробки даних, які надходять з акселерометра та гіроскопа використовується допоміжне ядро. ESP32 має пам'ять у розмірі 448 Кб, що дає змогу додатково зберігати отримані дані від GPS та акселерометра і це дає змогу не втрати дані у разі втрати з'єднання.

- робота GPS на частоті 19 Гц для більш точних вимірювань. Плата ESP32 дає в повному обсязі розкрити потенціал GPS модуля. В процесі налаштування було отримано максимальне значення, яке становить 19 Гц. Даний результат є задовільним, оскільки це краще ніж у більшості приладів представлених на ринку;

- програмне відключення GPS та акселерометра для економії батареї;
- можливість оновлення програмного забезпечення для апаратного засобу;
- високо частотна передача даних від акселерометра та гіроскопа на ESP32. Оскільки акселерометр та гіроскоп реалізовані в одному корпусі це дає змогу одночасної передачі даних на високій частоті;

- програмне забезпечення розроблено окремо для Android та iOS, що дозволить якісно оптимізувати роботу для обох операційних систем. Окрема розробка дозволить покращити безпеку та швидкодію програми, оскільки кожна операційна система має свої особливості.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

2.2. Архітектурні аспекти DRT

Під час розробки даного проекту було розроблено та випробувано декілька різних конфігурацій, а саме 3, що дало змогу детально проаналізувати і виділити їхні переваги та недоліки і обрати найкращу з них для її подальшого використання.

2.2.1. Пристрій на базі процесору Atmega 328p

При розробці першого пристрою були використані відносно дешеві комплектуючі. Пристрій складався з наступних компонентів[1]:

- процесор Atmega 328p;
- GPS Ublox NEO-6M-0-001;
- Bluetooth модуль HC-05;
- акселерометр, гіроскоп, термометр;

Переваги:

- низька вартість комплектуючих;
- швидка і легка заміна не працюючих елементів плати;
- можливість програмування мікроконтролера без використання сторонніх бібліотек;
- існує багато корпусів для процесора.

Недоліки:

- мікроконтролер можна легко пошкодити;
- процесор має лише 1 потік;
- живлення не вистачало для роботи окремих модулів;
- HC-05 не сумісний з iOS пристроями.

Atmega 328p – мікроконтролер сімейства AVR, як і всі інші має 8-бітний процесор і дозволяє виконувати більшість команд за один такт. Периферійні пристрої мікроконтролеру:

- два 8-бітних лічильника з модулями порівняння і дільником частоти;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

- 6 каналів PWM;
- програмований послідовний порт USART;
- послідовний інтерфейс SPI;
- інтерфейс I2C;
- внутрішня схема порівняння напруги;
- блок обробки переривань і пробудження при зміні напруги на виходах мікроконтролера;
- внутрішній калібрований генератор тактових імпульсів;

GPS Ublox NEO-6M – модуль використовується для отримання географічних координат поточного місцеположення антени приймача за допомогою GPS і їх передачі на контролер. Характеристики модулю:

- модуль плати: GY-GPS6MV2;
- GPS модуль: u-blox NEO-6M-0-001;
- батарейка для швидкого, холодного старту;
- передає координати в форматі: NMEA;
- є активна антена;
- швидкість підключення за замовчуванням по UART: 9600;

Основною причиною відмови від даної конфігурації було те що Bluetooth модуль HC-05 сумісний лише з Android пристроями, що унеможливлювало розробку програмного додатку для обох платформ. Все через те що даний модуль працює за протоколом Bluetooth 2, 3, але розробка програмного забезпечення для операційної системи iOS вимагає Bluetooth 4.0 BLE. Цю проблему можна вирішити лише заміною Bluetooth модуля. Також слід зауважити, що можна було б замінити Bluetooth на Wi-Fi, що дозволило б підключення обох платформ без додаткових труднощів і швидкість оновлення даних була б вищою. Але з інженерної точки зору та з точки зору зручності в даному проекті в користуванні було доцільніше використовувати Bluetooth, оскільки він має легше підключення і в цілому робота з ним значно зручніша.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		15

Також при розробці виникало досить багато інших складнощів, що не давало проекту рухатися далі. Середовище розробки вимагало спеціальне апаратне забезпечення для запису інформації на мікроконтролер, яке було досить незручним. Без додаткового живлення іншим модулям схеми зазвичай не вистачало живлення для стабільної та безперервної роботи. Оскільки існує багато корпусів для Atmega 328P, то в фінальній реалізації пристрою можна буде значно зекономити місця на монтажній платі.

Оскільки пристрій складався з великої кількості резисторів, транзисторів, діодів і так далі прорахувати собівартість такого пристрою було не можливо аж до самого закінчення проекту. Такий варіант проекту був найдорожчим по собівартості та найскладнішим для реалізації за решту, оскільки вимагав від розробника глибоких знань в мікроелектроніці. Дана конфігурація звичайно має своє право на існування, оскільки звільняє розробника від залежності від сторонніх бібліотек без яких не обійтись при підключенні того чи іншого модуля до плати.

Цю конфігурацію безумовно можна назвати найцікавішою з точки зору розробки, але після згорання одного процесору та блокування другого стало зрозуміло, що цей варіант займе надзвичайно багато часу, який потрібен на розробку та реалізацію.

Найкращим рішенням всіх труднощів, що виникли в процесі роботи стало – заміна процесора на плату типу Arduino, Raspberry Pi та інші, що і було зроблено. Таке рішення звільнило від розробки всієї схеми, підбору транзисторів, діодів, резисторів і так далі.

2.2.2. Пристрій під управлінням Arduino Nano

Наступна конфігурація, яка була зібрана і досліджена була під управлінням Arduino і складалася з наступних компонентів:

- репліка Arduino Nano;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		16

- GPS Ublox NEO-6M-0-001;
- Bluetooth модуль BT-05;
- гіроскоп, акселерометр, термометр;
- зарядний пристрій з micro-USB;
- акумуляторна батарея.

Переваги:

- зручне середовище розробки програмного забезпечення для Arduino;
- для програмування модулів використовується Micro USB;
- малі розміри плати Arduino Nano;
- вільний доступ до бібліотек для підключення модулів;
- захист від перенапруги;
- на вході всі модулі мають однакову напругу.

Недоліки:

- залежність від обов'язкових бібліотек для передачі даних від модулів до Arduino;
- одноядерний процесор;
- не можлива реалізація одночасної паралельної роботи 2 або більше модулів;

Було використано репліку Arduino Nano, а не оригінал, тому що вони різняться лише ціною та кількістю заводського браку, тож для цілей проекту його було більш ніж достатньо. Характеристика Arduino Nano[4][12]:

- вхідне живлення 7-12 В (рекомендоване);
- напруга живлення 5 В;
- 8 аналогових входів;
- максимальний струм цифрового виходу 50 мА;
- частота 16 МГц;
- ОЗУ 1 Кб;

Під час розробки та дослідженні цієї конфігурації було отримано багато позитивних результатів. Також стало відомо багато нюансів, які раніше були не відомі та не доступні для використання. Стало зрозуміло, що стандартна швидкість оновлення GPS модуля, який використовувався занадто мала для точних вимірювань.

Конфігурація з платою Arduino Nano дозволила визначити оптимальні апаратні рішення для даного пристрою і це дозволило зняти багать питань, щодо розробки. Це дало можливість рухатися далі. Стало зрозуміло, наскільки використання сторонніх плат економить час та ресурси розробника для створення продукту та реалізації його особливостей, якщо час, як в даному випадку є обмеженим.

З'явилась можливість сформулювати весь перелік вимог до проекту та зрозуміти можливості для їх реалізації. Однією з таких вимог стала організація паралельної роботи Bluetooth, GPS та акселерометра з гіроскопом. Без організації переривань реалізація цього стала не можливою, відповідно знизилась цінність проекту, як комерційного продукту.

Оскільки в даній конфігурації використовувався Bluetooth модуль BT-05, який працює з протоколом Bluetooth 4 стала можлива розробка програмного забезпечення під дві операційні системи Android та iOS.

Характеристика модулю BT-05:

- протокол зв'язку Bluetooth 4.0
- безпека ідентифікація та декодування;
- профіль Bluetooth Serial Port;
- частота GFSK;
- потужність відправки $\leq 4\text{дБм}$;
- потужність прийому $\leq 84\text{дБм}$.

Провівши налаштування GPS за допомогою програмного забезпечення, що надає виробник цього модулю, було отримане значення в 10Гц та згодом

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		18

стало зрозуміло, що можна отримати ще більшу частоту оновлення даних, а саме 19 Гц. Дана швидкість оновлення даних є більшою ніж у більшості конкурентів. Для такого типу пристрою така швидкість є більш ніж достатньою, оскільки даний проект є швидше для звичайного користувача ніж для професійних пілотів.

При підключенні GPS до плати та налаштуванні Arduino для отримання даних з GPS стало зрозуміло, що єдина доступна бібліотека, яка використовувалася для цього не здатна працювати на такій частоті. Використання інших бібліотек не є можливим, оскільки вони є старішими і не підходили під версію програмного забезпечення Arduino. Оскільки в такому разі цінність і доцільність розробки втрачається було вирішено змінити плату Arduino Nano на ESP32.

2.2.3. Пристрій під управлінням ESP32

Пристрій під управлінням плати на основі ESP32:

- плата з 2-во ядровим Bluetooth ESP32;
- GPS Ublox NEO-6M;
- Акселерометр, гіроскоп, термометр;
- зарядний пристрій з micro-USB;
- акумуляторна батарея.

Переваги:

- 2-во ядровий процесор;
- можливість організації паралельної роботи 2-ох модулів;
- собівартість апаратного забезпечення значно зменшилась;
- одночасна робота Bluetooth та Wi-Fi.

Недоліки:

- більший розмір плати;
- вищі температурні режими роботи;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		19

- складно купити в Україні, доставка плати триватиме 3-4 тижні.

За допомогою 2-го ядрового процесора нарешті з'явилась змога одночасно отримувати дані з 2 незалежних модулів, що було необхідно з самого початку роботи над проектом. Оновлення програмного забезпечення стало можливо передавати через Wi-Fi. За допомогою цього стало можливо після змін в програмному забезпеченні проводити перепрограмування плати віддалено, розмістивши плату з модулями біля вікна, оскільки GPS модуль погано отримує зв'язок із супутниками в будівлях з панельним або залізо-бетонним покриттям. З використанням Arduino було необхідно весь час бути біля вікна, щоб модуль мав зв'язок або після кожних змін ходити та ловити зв'язок по 3-5 хвилин.

Головною перевагою ESP32 є можливість відключати живлення від модулів, якщо відсутнє Bluetooth або Wi-Fi підключення, програмно без використання транзисторів, що дає змогу заощадити роботу акумуляторної батареї[14]. Оскільки ціна на ESP32 є така ж сама, як і репліки Arduino Nano, але ESP це і є Bluetooth/Wi-Fi модуль, то собівартість пристрою вдалось істотно зменшити, що позитивно вплинуло на комерційну складову даного проекту. Але більші розміри плати змушують переглянути розміщення всіх елементів на монтажній платі, оскільки розпайка на текстолітовій пластині відбувається вручну паяльником.

Для тестування різних бібліотек для роботи з GPS був куплений додатковий. Один працював з частотою 1Гц, а другий з частотою 19Гц. Це дозволило організувати «гарячу» заміну модулів, змінивши порт підключення програмним забезпеченням, що є дуже зручно в процесі розробки. Оскільки підбір бібліотек для ESP та GPS стало куди зручнішим та швидким процесом.

Головною проблемою даної конфігурації стали високі температурні режими роботи плати, що спонукало до організації достатньої терморегуляції корпусу, що змусило відмовитися на певний час від пило-волого захисту.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

Переконаний, що це питання буде вирішено в пізніших ревізіях пристрою, оскільки в реальних умовах прилад буде використовуватися в сніг, дощ, мороз, спеку і так далі.

Все одно ця конфігурація є найбільш вдалою на даному етапі за 2 попередні. Можливо в майбутньому доведеться відмовитися від даної плати на більш потужну, використати інший GPS модуль. Але станом на сьогодні не має в цьому потреби. Більш потужні плати є значно дорожчими та громіздкими, що робить пристрій менш зручнішим та привабливим для покупки.

2.3 Проектування та організація структури DRT.

В основу всього проекту було покладено GPS модуль Ublox NEO-6M, що є пристроєм початкового класу, але достатнім для вимог даного проекту. Модуль стандартно працює на частоті в 1 Гц. Компанія Ublox надає програмне забезпечення за допомогою якого можна здійснити налаштування пристрою на необхідну для використання швидкість. Щоб налаштувати прилад необхідна підключити GPS до комп'ютера за допомогою USB-UART адаптера. Не зважаючи на те що даний модуль не є професіональним, він здатний працювати на частоті в 19 Гц. Така частота є не найшвидшою у порівнянні з вже існуючими рішеннями. Ця частота є краще на 90% за більшість конкурентів та уступає лише одному приладу. GPS модуль оснащений антеною в керамічному корпусі з можливістю від'єднання від плати. На самій схемі та антені передбачені місця для розпайки антени на платі. На корпусі пристрою передбачені технологічні отвори для проведення проводів або для кріплення схеми до корпусу пристрою болтами. На платі присутній світловий діод для індикації зв'язку з супутниками. Якщо з'єднання з супутником не встановлено – індикатор не працюватиме.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		21

Керуючим пристроєм є плата ESP32 з вбудованим Wi-Fi/Bluetooth модулем. Схема має 18 цифрових контактів для зчитування та запису інформації на зовнішні модулі. Запис даних для управління всією системою відбувається через Micro-USB роз'єм. Плата має вбудований модуль Wi-Fi та Bluetooth зі своєю антеною, місце для підключення зовнішньої антени не передбачене. При проектуванні була вибрана саме ця плата в якості керуючого пристрою, оскільки розробка програмного забезпечення відбувається за допомогою середовища розробки Arduino та апаратно оснащений 2-во ядерним процесором, що дозволяю легко та зручно організувати паралельну роботу зовнішніх модулів не прибігаючи до переривань. На платі є 2 контакти для землі, що з'єднанні між собою. Передбачений контакт для зовнішнього живлення. Недоліком плати є її великі розміри та високі температурні режими роботи Wi-Fi/Bluetooth модуля. На корпусі пристрою передбачені технологічні отвори для проведення проводів або для кріплення схеми до корпусу пристрою болтами. Завдяки використанню цього пристрою з'являється можливість відключати живлення на модулі, якщо відсутнє з'єднання з користувачем. Це дозволяє економити заряд акумуляторної батареї та цим продовжує час автономної роботи всього приладу, що є дуже зручно з точки зору використання.

Для вимірювання прискорення та зміни висоти використовується 3-х осьовий акселерометр та 3-х осьовий гіроскоп, який оснований на базі плати MPU6050. Для якісного аналізу вимірювань слід мати інформацію про зміну висоти, оскільки це може бути ключовим фактором у вимірюваннях. Акселерометр необхідний для того, щоб вимірювати навантаження під час прискорення та впродовж всього заїзду. Дані про навантаження не завжди є необхідними для користувача. Необхідність в цих даних виникає після змін в потужності автомобіля. Це дає змогу аналізувати про зміни в динаміці автомобіля. Модуль також оснащений термометром, але необхідності в його використанні нема, тому що він буде відображати температуру в середині

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		22

пристрою, що не є потрібним для користувача, але на етапі проектування є необхідним для розміщення модулів в корпусі. Його використання дозволяє аналізувати яка температура в середині пристрою і розміщати модулі так, щоб прилади не перегрівалися та не виходили з ладу. Оскільки гіроскоп та акселерометр є 3-х осьовими, то при розробці можна визначати як буде розміщено модуль в платі, горизонтально або вертикально та програмно це зазначити. На модулі є 8 контактів серед яких лінія даних I2C та лінія синхроімпульсів I2C. Також передбачений контакт для налаштованих переривань та 2 додаткових I2C інтерфейси для підключення зовнішнього магнітометра.

Для організації автономної роботи пристрою використовується акумуляторна батарея Li-Po типу на 600 мАг на 3,7 вольт. Був вибраний саме цей тип батареї, оскільки такі акумуляторні батареї в менший розмір вміщують більший об'єм заряду. В якості зарядного пристрою та для контролю від глибокого розряду використовується зарядний пристрій з захистом. В прилад вбудований micro-USB роз'єм для заряджання. Використовується саме такий формат входу, тому що він є найбільш поширеним і терміново знайти провід для зарядки не стане проблемою. Можливо заряджати пристрій через передбачені для цього контакти. На платі передбачені контакти для з'єднання з керуючим пристроєм для передачі живлення та про стан заряду батареї. На схемі є 2 світлових діода, які працюють лише під час заряджання батареї і сповіщають про повний заряд. Прилад має досить компактний роз'єм і на монтажній платі може бути розміщений під керуючий пристрій. Недоліком плати є високий температурний режим під час заряджання.

Після всіх тестувань та налаштувань всі модулі буде розміщено та вручну розпаяно на текстолітовій пластині з отворами покритими міддю. Кінцеву плату буде поміщено в корпус.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		23

2.4 Технологія пристрою.

Під час проектування було задіяно досить багато технологій, що позитивно вплинуло на прилад. Це були як програмні, так і апаратні рішення. Головна складність виникла в тому, що певні рішення перешкоджали роботі інших. Для вирішення таких питань проект доводилося навіть змінювати концептуально.

Технологією з точки зору апаратного рішення можна назвати:

- використання 3-х осьових гіроскопа та акселерометра для зручного та компактного розміщення в корпусі пристрою;

У даному проекті для корегування даних про швидкість і для вираховування перенавантаження, яке відчуває користувач під час гальмування був використаний сучасний 3-х осьовий акселерометр та гіроскоп MPU6050 GY-521, який керується протоколом I2C. Даний пристрій має наступні діапазони акселерометра та гіроскопа відповідно (+ 250 500 1000 2000 */с) та ($\pm 2 \pm 4 \pm 8 \pm 16$ g). Даний модуль має наступні розміри 2 см на 1.6 см, тому цей модуль є досить компактним. MPU6050 має 8 входів із яких використовуються в даному проекті лише 4.

- використання сучасного micro-USB порту для обміну даних на етапі розробки та для зарядження пристрою;

У плату контролер ESP32 влаштований micro-USB. Даний роз'єм призначений для обміну інформацією між даним пристроєм і зовнішнім ПК. Micro-USB слугує не лише для зарядки, а і для синхронізації. Даний порт має 5 “ніжок”: одна плюсова на 5 вольт, а одна мінусова. Ще дві відповідають за синхронізації, тобто за можливість запису та вивантаження різних даних. В даному проекті ці дві “ніжки” Data+ та Data- використовуються для запису коду для подальшого керування всією платою. Остання “ніжка” використовується для швидкої зарядки пристрою. Якщо вона зламана то

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		24

зарядження пристрою виконуватися не буде. В основному micro-USB використовується в даній розробці для обміну даних на етапі розробці.

- використання GPS на частоті в 19Гц, за допомогою цього можна отримати досить точні і коректні дані про вимірювання;

В даному пристрої використовується GPS модуль NEO-6М. Початкова частота оновлень даного модуля була 1Гц. GPS використовується для отримання швидкості та географічних координат теперішнього положення антени приймача. В ході розробки було виявлено, що початкової швидкості отримання даних не достатньо для відображення чіткої картини замірів. Тому був проведений поступовий “розгін” пристрою до 19Гц з кроком в 1Гц. Така частота дозволила отримувати набагато чіткіші і коректніші дані про швидкість і місцеположення пристрою.

- використання Li-Pol батареї, яка дозволила зекономити місце та отримати більшу ємність батареї:

Даний пристрій в якості живлення використовує акумуляторну літій-полімерну батарею. Літій-полімерний акумулятор – це удосконалена конструкція літій-іонного акумулятора. В якості електроліту використовується полімерний матеріал. До переваг використання даного типу батареї можна віднести:

- велику густину енергії на одиницю маси;
- низький саморозряд,
- товщина елементів від 1 мм;
- можливість отримувати дуже гнучкі форми;
- слабо виражений ефект пам’яті;
- незначний перепад напруги по мірі розрядження;
- діапазон робочих температур літій-полімерних акумуляторів досить широкий: від -20 до +40°C .

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		25

Розміри батареї досить компактні тому цей акумулятор живлення ідеально вписується в наш пристрій.

- використання керуючої плати з 2-во ядерним процесором для запобігання переривань:

За основу пристрою була взята плата ESP32, яка має 32-бітний 2-во ядерний мікропроцесор Tensilica Xtensa LX6, який в свою чергу працює на частоті 160 і 240 МГц. Це дало змогу значно підвищити швидкодію пристрою, оскільки програмне забезпечення для апаратного забезпечення написане з використанням багатопотоковості для одночасної роботи з GPS та акселерометром і гіроскопом. Якби замість даного мікропроцесору був використаний якийсь інший одноядерний процесор ефективність, швидкодія і точність вимірювань упала в 2 рази. За рахунок того що вдалося організувати роботу пристрою на 2-во ядерному процесорі не знадобилось використовувати переривання, що на додаток ще підвищили швидкодію пристрою і позитивно сприяло на результати;

- захист від перенапруги шляхом використання додаткових захисних модулів:

В даному проекті використовується літій-полімерна акумуляторна батарея. Оскільки акумулятори даного типу є пожежонебезпечними при перезаряді і/або перегріві то додатково був введений допоміжний модуль контролю заряду. Контролер заряду має наступні 4 входи та виходи: Input+, Input-, Output+, Output-. Даний контролер дозволяє вести контроль за зарядом літій-полімерної батареї, відключати зарядку при надмірному заряді акумуляторної батареї. До того ж стабілізує напругу на всьому пристрої в цілому за рахунок подачі живлення на VIN вхід ESP32. Інші функції контролера:

- компенсація різниць потоків енергії при постачанні споживача енергією одночасно з зарядкою акумулятора;

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		26

- вимірювання температури для аварійного відключення зарядки на холоді чи перегріві (для запобігання псування акумулятору);
- замір тиску за допомогою датчика тиску для аварійного відключення зарядки при витоці газів (для запобігання вибухів і витоків);

Контроль заряду батареї виконуються за наступним алгоритмом:

- вимір часу з початку зарядки;
- замір напруги і струму на вході акумулятора;
- зміна величин струму і напруги заряду в залежності від замірених значень;
- повторення циклів заряд/розряд (для відновлення ємності акумулятору);
- заряд до 90% ємності акумулятору (для підвищення терміну служби).

З точки зору ПЗ можна виділити наступні технології:

- фільтрування даних отриманих від акселерометра та гіроскопа шляхом математичних обчислень:

для фільтрування даних отриманих від акселерометра та гіроскопа був використаний спеціальний фільтр, який дозволив згладити дані, оскільки початкові дані які надходять від модуля не є коректними. Фільтр дозволяє зробити роботу пристрою більш стабільною за рахунок фільтрування фонових шумів вимірювань, а також може відфільтрувати помилкові спрацьовування датчиків. Використання даного фільтру дозволило отримати чіткіші дані на які не впливають сторонні чинники.

- обробка даних отриманих з GPS та швидке переведення їх в одиниці швидкості:

Дані які надходять з GPS пристрою мають формат NMEA. Всі команди і повідомлення передаються в текстовому ASCII-виді. Повідомлення, яке відноситься до GPS-приймача починається з \$GP, в кінці рядка повідомлення

мають бути символи. Дані від GPS відправляються на плату ESP32, де зберігаються в відповідний масив до надходження символу кінця повідомлення. Потім дані розділяються і виділяються потрібні із них, а саме дані про швидкість, положення і так далі.

- зберігання даних та подальше їх архівування з метою економії пам'яті пристрою користувача:

Дані які надходять на пристрій проходять попередню обробку, зберігаються в відповідні змінні для тимчасового відображення поточних даних. Потім дані проходять всі необхідні перевірки, архівуються LZW алгоритмом та локально зберігаються для економії пам'яті пристрою користувача. LZW алгоритм – це універсальний алгоритм стиснення даних без втрат. Алгоритм був розроблений таким чином, щоб його можна було легко реалізувати як програмно так і апаратно. Даний алгоритм при кодуванні повідомлення динамічно створює словник фраз: визначеним послідовностям символів ставляться у відповідність групи бітів фіксованої довжини.

- сортування отриманих даних на дійсні та не дійсні з метою відображення якості отриманого результату:

Результати, які отримує пристрій користувача зберігаються у пам'яті і відображаються у відповідних фрагментах додатку. Оскільки дані постійно оновлюються то був введений контроль на актуальність даних. Якщо відображені дані не актуальні на даний момент то вони замінюються на більш актуальні, які прийшли останніми. Також ведеться контроль на відкидання помилкових і не потрібних даних.

- відключення модулів GPS, акселерометра, гіроскопа тоді, коли нема з'єднання з гаджетом користувача з метою економії заряду акумуляторної батареї:

Після того як з'єднання між телефон користувача та пристроєм встановлено програмно подається живлення на модулі GPS та акселерометр з гіроскопом. І

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		28

відповідно, якщо зв'язок з пристроєм втрачається подається відповідний сигнал після якого на дані пристрої подається слабка напруга, яка не дає модулям повністю відключитися. Це зроблено для того щоб при наступному підключенні не витрачався додатковий час на встановлення з'єднання між модулями та встановлення з'єднання між GPS модулем та супутником. Такий підхід майже не витрачає ресурси батареї і тому є оптимальним.

- система безпеки запобігання MITM атак для попередження втрати результатів:

MITM атаки – вид атаки в криптографії, коли зловмисник таємно ретранслює і при необхідності змінює зв'язок між двома сторонами, які вважають, що вони безпосередньо спілкуються один з одним. Є методом копроментатії каналу зв'язку, при якому зловмисник, підключившись до каналу між контрагентами, здійснює втручання в протокол передачі, видаляючи або змінюючи інформацію. Для запобігання MITM атак в даному проекті була запроваджена система безпеки і дані які відправляються від пристрою до додатку без завад (у захищеному режимі). Також є додаткова перевірка сеансу зв'язку по Bluetooth/Wi-Fi каналу. Тим самим ця система безпеки забезпечує безпомилкову передачу даних для запобігання невірних результатів.

2.5. Типова реалізація. Приклад розробки

У ході розробки було випробувано 3 різних конфігурації, що дозволило знайти оптимальну реалізацію даного пристрою. Тож розроблений прилад складався з наступних складових:

- акселерометр, гіроскоп, термометр;
- плата з 2-во ядровим Bluetooth ESP32;
- GPS Ublox NEO-6M;
- акумуляторна батарея.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		29

У даному пристрої була використаний модуль GY-521 MPU-6050, який включає в себе акселерометр, гіроскоп та термометр. Дає змогу визначити положення і переміщення приладу в просторі: кути крену, диференту орієнтуючись по вектору сили тяжіння і швидкості обертання. Вимірює температуру. При переміщенні визначає лінійне прискорення і кутову швидкість по трьох осях, що дає повну картину стану. Основа модуля - мікросхема MPU-6050. Містить два пристрої акселерометр і гіроскоп. Їх дані проходять попередню обробку і передаються по послідовному інтерфейсу I2C в мікроконтролер. Модуль 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050 є електронним аналогом нашого вестибулярного апарату. Завдяки вестибулярному апарату або схожим органу живі істоти відчують напрямок тяжіння. Вестибулярний апарат на підсвідомому рівні дозволяє нам тримати рівновагу.

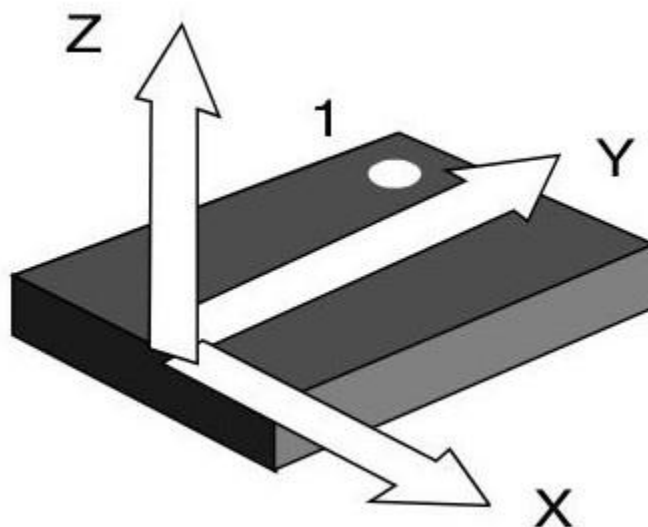


Рисунок 2.1 – Осі акселерометра

3-ох осьовий акселерометр "відчуває" проекції прискорення на осі X, Y і Z. Якщо прилад розміщений строго горизонтально і не рухається то проекції прискорення сили тяжіння на осі X і Y дорівнюють нулю. Сила тяжіння сприймається тільки чутливими елементами вертикальної осі Z. Час від часу в

стані спокою проводять перевірку і калібрування акселерометра. Під час руху об'єкт постійно то прискорюється, то сповільнюється. Ідеально рівномірного руху не існує. Це і дозволяє використовувати акселерометр не тільки для визначення положення об'єкта, а й для визначення динамічних параметрів при русі. Акселерометр реєструє суму прискорення при русі і гравітацію. Якщо від всіх елементів акселерометра по осях X, Y, Z надходять свідчення близькі до нуля, значить, двигун вимкнений.

3-ох осьовий гіроскоп – це датчик повороту об'єкту, який дозволяє вираховувати кути поворотів по осям X, Y, Z завдяки визначенню кутових швидкостей.

Для правильної обробки параметрів руху і вірного розпізнавання динамічних параметрів застосували спільно акселерометр і гіроскоп. При використанні гіроскопа в даному випадку він визначить кут повороту і дасть можливість правильно інтерпретувати дані акселерометра. Використання гіроскопа без акселерометра неможливо через особливості математики гіроскопа, яка призводить до накопичення похибки. Спеціальна математика дозволяє об'єднати обробку даних від обох датчиків.

Для точного відстеження рухів передбачена можливість запису в пам'ять MPU-6050 актуальних меж вимірювань. Дані можна зчитувати з регістрів зберігання або буфера FIFO розміром 1024 байт. Мікросхема MPU-6050 може працювати в режимі майстер на шині I2C для контактів XDA і XCL. Містить АЦП 16 біт. Мікросхема MPU-6050 містить більше 100 регістрів.

Адреса мікросхеми може бути двох значень (без біта читання / запису) в залежності від стану виведення AD0 модуля 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050:

- 0x68, якщо AD0 з'єднаний із загальним проводом;
- 0x69, якщо AD0 з'єднаний з потенціалом логічна "1".

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		31

На початку програми відбувається калібрування за першими 10 даними для отримання постійних похибок (зсувів) від датчиків. Зміщення віднімаємо з необроблених значень датчика до перетворення даних в кути.

Для обчислення орієнтації акселерометром, ми спираємося на той факт, що відбувається постійне гравітаційне тяжіння 1 g. Якщо немає додаткових сил, що діють на акселерометр, то виявлена величина прискорення буде 1 g. Ділення сирих даних гіроскопа на 131 дає кутову швидкість в градусах в секунду. 131 - коефіцієнт чутливості гіроскопа в заданому режимі 250 град / с. Оскільки у нього АЦП 16 біт, то модуль максимального необробленого значення дорівнює 32767. Тепер $32767/250 = 131$ умовних одиниць на градус в секунду. Тобто, якщо необроблене значення дорівнює 131, то кутова швидкість дорівнює 1 градус в секунду. За допомогою цих даних можна отримати положення об'єкта. Для цього миттєве значення кутової швидкості помножимо на проміжок часу між опитуваннями датчика гіроскопа. Наприклад, розширення 2000 градусів в секунду, час між опитуваннями 0,1 с, значення миттєвої швидкості 210, значить $210 * 0,1 = 21$ - за цей час стався поворот на 21 градус. Далі кожне отримане значення потрібно скласти з попереднім.

Як було описано вище дані акселерометра і гіроскопа схильні до систематичних помилок. Акселерометр забезпечує отримання точних даних в довгостроковій перспективі, але дає сильний шум при розгляді за короткий проміжок часу. Гіроскоп забезпечує отримання точних даних про зміну орієнтації за короткий проміжок часу, але при інтеграції даних відбувається дрейф.

Вирішення цих проблем полягає у використанні даних акселерометра і гіроскопа разом таким чином, щоб помилки взаємно знищувалися.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		32

Наступний модуль який був використаний при реалізації пристрою є плата з 2-во ядровим Bluetooth ESP32. Дана плата була обрана через можливість швидкої і безпомилкової передачі даних між пристроєм і додатком.

ESP32 - потужний високопродуктивний модуль загального призначення з вбудованим мікроконтролерним ядром і з підтримкою стандартів зв'язку Wi-Fi + BT + BLE, призначений для широкого ряду додатків з економічним енергоспоживанням. Модуль побудований на базі чіпсета ESP32-D0WDQ6. Чіпсет створювався, виходячи з міркувань максимально легкого проектування і адаптації до умов застосування. Чіпсет містить 2-во ядерний процесор з тактовою частотою від 80 МГц до 240 МГц. У користувача є можливість повністю відключити вбудоване CPU і залишити працювати тільки співпроцесор для постійного моніторингу периферії на предмет перевищення порога сигналами зовнішніх датчиків. ESP32 інтегрує дуже великий набір вбудованої периферії, включаючи ємнісні сенсори торкання, малошумні підсилювачі, інтерфейс SD-карти, Ethernet, високошвидкісний SPI, UART, I2S і I2C.

Вбудована підтримка Bluetooth, Bluetooth LE і Wi-Fi гарантує високу ефективність застосування модуля з великим доробком на майбутнє: Wi-Fi дозволяє говорити про високу ефективність застосування в задачах, де відбувається підключення до мережі Інтернет через Wi-Fi - роутер, завдяки Bluetooth користувач може підключати модуль до смартфона або будь-якого іншого призначеного для користувача пристрою. Струм сплячого режиму ESP32 становить менше 5 мкА, що робить його оптимальним рішенням для пристроїв з батарейним харчуванням. ESP32 підтримує швидкість передачі даних до 150 Мбіт / с, і підводиться до антени вихідну потужність 20.5 дБм. Таким чином, чіп являє собою лідируючу в галузі специфікацію параметрів і чудову продуктивність процесорного ядра, контролера бездротового зв'язку,

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		33

високу дальність зв'язку, оптимальне енергоспоживання, широкий ряд комунікаційних можливостей[9]. Операційна система, яка використовується в складі ядра ESP32 - freeRTOS з підтримкою технологій LwIP і TLS 1.2 з апаратним прискоренням. Крім усього іншого підтримується безпечне (шифрування) оновлення прошивки по радіоканалу (OTA) . Так що, вбудоване ПО кінцевого пристрою завжди може підтримуватися в актуальному стані.

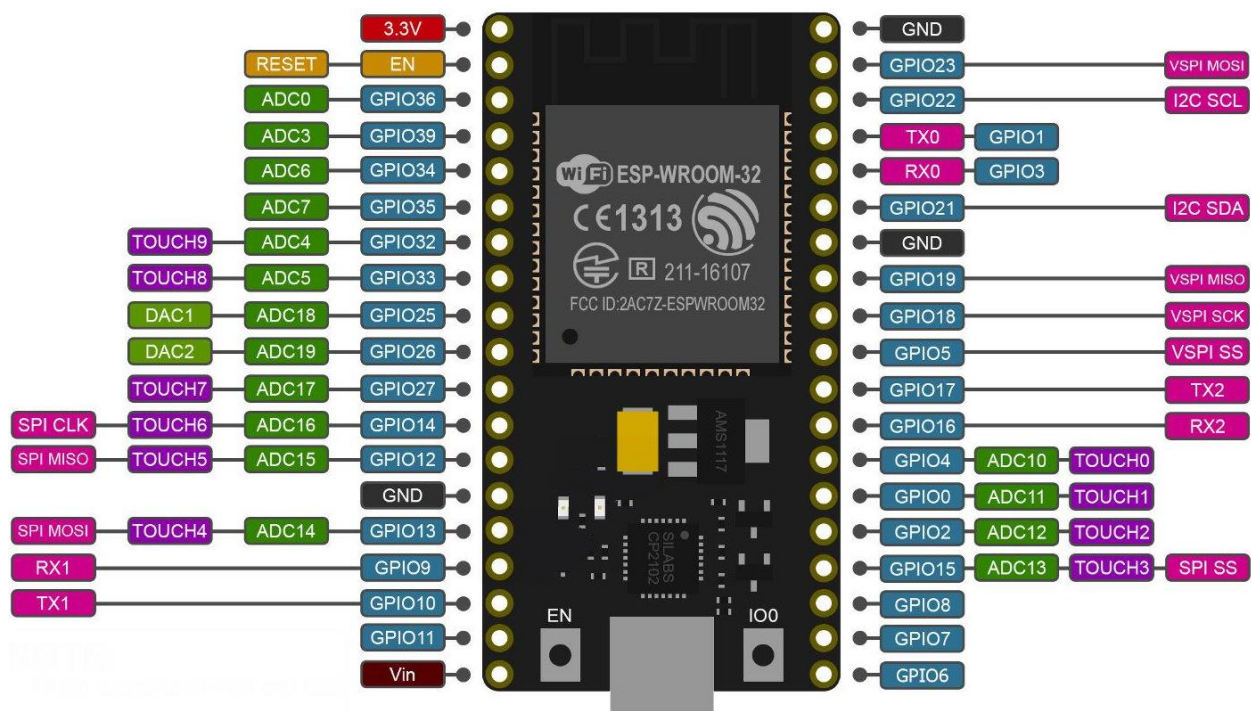


Рисунок 2.2 – Піни модуля ESP32

До портів GPIO 0, 4, 2, 15, 13, 12, 14, 27, 33 і 32 підключені сенсорні виходи. Вони можуть використовуватися для виведення ESP32 з глибокого сну. Вони фіксують будь-яку зміну електричного заряду[8].

Порти з 34 по 39 використовуються тільки для введення інформації. На них відсутні додаткові резистори, тому їх не можна використовувати як виходи.

Також ESP32 має 10 внутрішніх ємнісних сенсорних датчиків. Вони можуть відчувати зміни у всьому, що містить електричний заряд, навіть на людській шкірі. Таким чином, вони можуть виявляти зміни, викликані

дотиком до GPIO пальцем. Ці виходи легко працюють з ємнісними панелями і замінюють механічні кнопки. Ємнісні сенсорні виходи також можуть використовуватися для пробудження ESP32 від глибокого сну. Ці датчики підключені до наступних GPIO: T0 (GPIO 4), T1 (GPIO 0), T2 (GPIO 2), T3 (GPIO 15), T4 (GPIO 13), T5 (GPIO 12), T6 (GPIO 14), T7 (GPIO 27), T8 (GPIO 33), T9 (GPIO 32).

Для живлення у схемі використовується літій-полімерна акумуляторна батарея та модуль контролю заряду для запобігання перезаряджання батарея та контролювання напругу на вході схеми.

Літій-полімерна батарея – це удосконалена конструкція літій-іонного акумулятору. В якості електроліту використовується полімерний матеріал. Використовується в мобільних телефонах, цифровій техніці, радіокерованих моделях і так далі.

Літій-полімерні акумулятори при однаковій вазі перевершують по енергоемності нікель-кадмієві батареї в 4-5 разів, а нікель-металогідридні в 3-4 рази. Кількість робочих циклів 500-600.

Зарядження акумуляторів виконується за допомогою досить простого алгоритму – зарядження від джерела постійної напруги 4.2 В на елемент з обмеженням струму в 1С. Заряд вважається завершеним, коли струм впаде до позначки в 0.1 – 0.2 С. Після переходу в режим стабілізації напруги при струмі в 1С акумулятор набирає приблизно 70-80 % ємності. Для повної зарядки необхідно приблизно дві години.

Головним модулем розробленого пристрою є GPS Ublox NEO-6М. Насамперед потрібно розповісти про GPS.

Отже, GPS (Global Positioning System – система глобального позиціонування) – супутникова система навігації, яка забезпечує вимірювання відстані, часу і визначає місце розташування у всесвітній системі координат

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		35

WGS 84. Дозволяє майже при будь-якій погоді визначати місце розташування в будь-якому місці Землі (виключаючи приполярні області) і навколоземного космічного простору. Система розроблена, реалізована і експлуатується Міністерством оборони США, при цьому в даний час доступна для використання для цивільних цілей - потрібен тільки навігатор або інший апарат (наприклад, смартфон) з GPS-приймачем.

Основний принцип використання системи - визначення місця розташування шляхом вимірювання моментів часу прийому синхронізованого сигналу від навігаційних супутників антеною споживача. Для визначення тривимірних координат GPS-приймача потрібно мати чотири рівняння: «відстань дорівнює добутку швидкості світла на різницю моментів прийому сигналу споживачем і моменту його синхронного випромінювання від супутників».

GPS складається з трьох основних сегментів: космічного, керуючого і призначеного для користувача. Супутники GPS транслюють сигнал з космосу, і все приймачі GPS використовують цей сигнал для обчислення свого положення в просторі за трьома координатам в режимі реального часу.

Космічний сегмент складається з 32 супутників, що обертаються на середньої орбіті Землі.

Станом на 8 вересня 2018 року використовуються за цільовим призначенням 31 космічний апарат (КА). На етапі введення в систему 0 КА, виведені на техобслуговування 1 КА.

Керуючий сегмент являє собою головну керуючу станцію і кілька додаткових станцій, а також наземні антени і станції моніторингу, ресурси деяких зі згаданих є спільними з іншими проектами.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		36

Призначений для користувача сегмент представлений приймачами GPS, які перебувають у віданні державних інститутів, і сотнями мільйонів приймальних пристроїв, власниками яких є звичайні користувачі.

Попри те, що проекти побудови GPS-систем впроваджувались військовими відомствами, зараз, окрім приймачів спеціального призначення, випускаються прилади, вмонтовані в різноманітну дрібну техніку: наручні годинники, мобільні телефони, ручні радіостанції, портативні комп'ютери та фотоапарати, за допомогою яких можна орієнтуватися на місцевості або фіксувати місцезнаходження користувача. Їх використовують альпіністи, рятівники, туристи.

Споживачам також пропонуються різні пристрої й програмні продукти, котрі дозволяють:

- бачити своє місцезнаходження на електронній карті;
- прокладати маршрути з урахуванням дорожніх знаків, дозволених поворотів і навіть заторів;
- шукати на карті конкретні будинки й вулиці, визначні пам'ятки, кафе, лікарні, автозаправки та інші об'єкти інфраструктури.

Окремий клас пристроїв, GPS-трекери, призначений для отримання інформації про рух обладнаних ними автомобілів або інших рухомих об'єктів у пункті спостереження. Використання GPS-трекерів дозволяє будувати диспетчерські системи спостереження та управління рухом, системи GPS-моніторингу транспорту.

Системи GPS-моніторингу рухомих об'єктів бувають двох видів:

- системи offline;
- системи online;

Системи Offline — використовують принцип «чорного ящика», який фіксує події, прив'язуючи кожен з них до географічних координат і реального часу.

Після прибуття об'єкта на базу вся інформація з такого «чорного ящика» зчитується, розшифровується й стає доступною для детального аналізу та агрегації.

Переваги підходу:

- велика кількість різномірної інформації, яка збирається пристроєм (обмежена тільки об'ємом власної пам'яті пристрою).
- відсутність абонентської плати за передачу даних (передача здійснюється або через фізичне підключення пристрою до комп'ютера диспетчера, або через локальні бездротові мережі).

Недоліки:

- інформація доступна тільки після прибуття об'єкта на базу.

Системи Online — використовують принцип радіомаяків — інформація про місцезнаходження не фіксується в пам'яті приладів, а передається в реальному часі диспетчеру. Каналами передачі даних в такому разі може виступати GSM, SMS-повідомлення, GPRS, радіозв'язок, мережа бездротових точок доступу.

Переваги системи:

- Інформація про переміщення та стан об'єкта доступні диспетчеру в реальному часі (можлива затримка обумовлена способом комунікацій).
- Час відгуку пристрою моніторингу залежить від каналу передачі даних (завжди є можливість підібрати компромісний варіант за співвідношенням ціна/оперативність).

Недоліки:

- Необхідність оплачувати послуги зв'язку (розмір плати змінюється від кількох центів за SMS-повідомлення в межах мережі свого стільникового оператора до декількох доларів за GPRS-з'єднання в роумінгу).

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		38

Модуль GPS Ublox NEO-6M використовується для отримання географічних координат теперішнього положення антени приймача за допомогою GPS і їх передачі на контролер. Модуль може використовуватися в пристроях на базі Arduino, AVR, PIC, ARM.

Для використання GPS модулю потрібно спочатку його підключити до живлення. Потім потрібно очікувати поки модуль не встановить з'єднання зі супутником.

Для закріплення модулю на плоскій поверхні на корпусі є 4 отвори.

Управління відбувається за допомогою контролеру по інтерфейсам UART, SPI, DDC, ІС. На платі є один UART інтерфейс для підключення живлення і вводу та виводу інформації.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		39

3. РЕАЛІЗАЦІЯ ПРОЕКТУ

3.1 Вибір схеми моделі

За основу даного проекту була обрана схема з найкращими показниками із усіх протестованих. Схема зображена на рис. 3.1 є оптимальною для поставленої задачі.

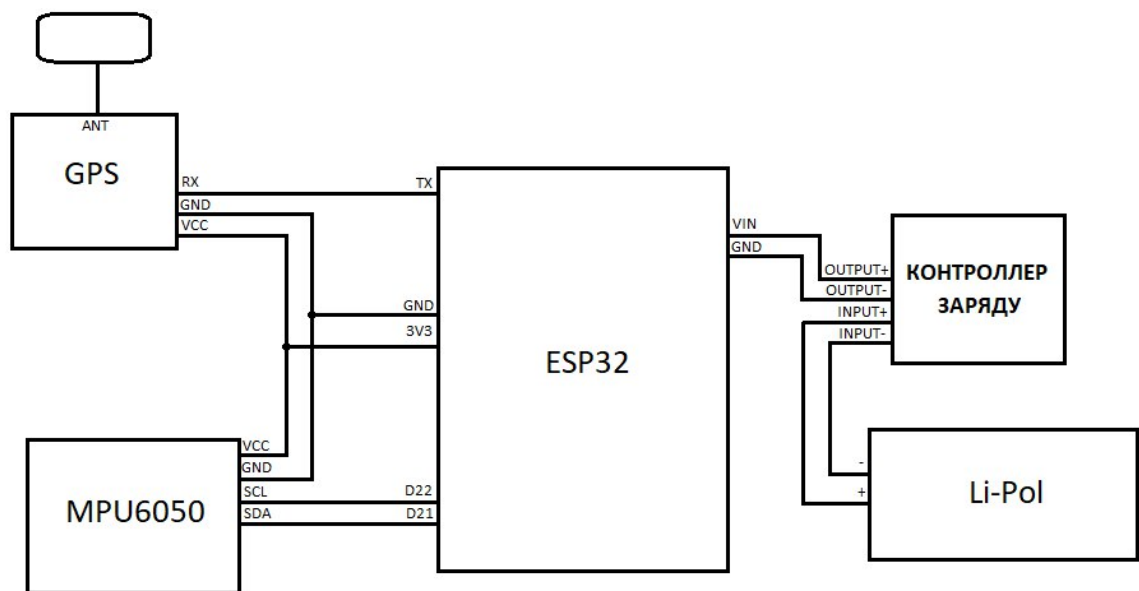


Рисунок 3.1 - Схема модель дипломного проекту

Вбудований Wi-Fi/Bluetooth модуль в плату ESP32 дозволяє зберегти основні входи та виходи на схемі для більш важливих модулів, а саме для підключення GPS до входу TX на платі ESP32. Це дозволило використовувати можливості GPS на повну.

Переваги використання літій-полімерної батареї:

- велика густина енергії на одиницю маси;
- низький саморозряд;
- товщина елементів від 1 мм;
- можливість отримувати дуже гнучкі форми;
- слабо виражений ефект пам'яті;

- незначний перепад напруги по мірі розрядження;
- діапазон робочих температур літій-полімерних акумуляторів досить широкий: від -20 до +40°C .

Із літій-полімерної батареї виходить 2 мідні проводи живлення, що мають бути підключені до проміжного пристрою керування напруги або напряму до плати, що не є добре. Підключення напряму до плати може бути небезпечним через те що у випадку перенапруги може “згоріти” весь пристрій або батарея критично розрядиться і втратить свою ефективність, що негативно вплине на подальше користування всім пристроєм. Акумуляторна батарея може зберігати довго свій заряд за рахунок своїх властивостей і тим самим схема може довго залишатися робочою.

Контролер заряду має наступні входи та виходи:

- Input+ - вхідне позитивне підключення;
- Input- - вхідне негативне підключення;
- Output+ - вихідне позитивне підключення;
- Output- - вихідне негативне підключення;

Контролер заряду підключений входами Input+, Input- до входів батареї + та – відповідно. Даний контролер дозволяє запобігати перевантаженню схеми та контролює заряд батареї. Також він підключений на входу VIN на платі ESP32, що дозволяє якісно розподіляти вхідну напругу на всі модулі в схемі.

В Модулі MPU6050 були використані наступні входи та виходи:

- SCL – лінія синхроімпульсів I2C;
- SDA – лінія даних I2C;
- GND – земля;
- VCC – позитивний контакт живлення;
- INT – налаштовуване переривання;

- AD0 – I2C адреса; за замовчуванням AD0 підтягнутий до землі, тому адреса пристрою – 0x68, якщо з'єднати AD0 до контактів живлення то адреса зміниться на 0x69;
- XCL, XDA – додатковий I2C інтерфейс для підключення зовнішнього магнітометра.

Входи SCL та SDA підключаються до спеціальних входів D22 та D21 відповідно. Таке підключення не є випадковим, адже саме така конфігурація дозволяє безперешкодну та швидку передачу даних від датчика акселерометра та гіроскопа до керуючого пристрою. Входи D22 та D21 дозволяють швидкий прийом даних та подальшу їх обробку.

Модуль ESP32, який живиться від контролера заряду, живить всі інші модулі схеми напругою 3,7 В. Така напруга була обрана для запобігання перевантаження схеми і дозволяє всім елементам працювати коректно і без перебоїв.

Також у схемі був використаний GPS модуль Ublox NEO-6m. На платі модуля є один UART інтерфейс для підключення живлення і вводу та виведення інформації. Визначення контактів:

- VCC (напруга живлення);
- RX (вхідні дані);
- TX (вихідні дані);
- GND (загальний контакт).

Даний GPS модуль підключається до супутників і отримує всю потрібну нам інформацію про швидкість, положення об'єкта і так далі. Далі він передає всю отриману інформацію на ESP32, де виконуються розділення отриманих даних і збереження їх у відповідні комірки пам'яті.

GPS модуль був підключений до входів RX, TX

3.2. Вибір інструментального програмного забезпечення

Проаналізувавши всі доступні середовища розробки для написання програм під операційну систему Android був обраний один беззаперечний лідер в цьому напрямку розробки і ним став Android Studio IDE.

Android Studio – це інтегроване середовище розробки (IDE) для роботи з платформою Android.

Android Studio заснована на програмному забезпеченні IntelliJ IDEA – офіційне середовище розробки Android додатків.

Рішення для Android розробляються в Android Studio з використанням Java, C++, або Kotlin[2][6]. В основі робочого процесу Android Studio закладений концепт безперервної інтеграції, який дозволяє одразу знаходити присутні помилки. Продовжена перевірка коду забезпечує можливістю ефективного зворотного зв'язку з розробниками. Така опція дозволяє швидше опубліковувати версію мобільного додатку в Google Play App Store.

За допомогою засобів оцінки продуктивності визначається стан файлу з пакетом прикладних програм[3]. Візуалізація графіки дає можливість дізнатися чи підходить додаток орієнтиру Google в 16 мс. За допомогою інструменту для візуалізації пам'яті є можливість дізнатися, коли додаток буде використовувати занадто багато оперативної пам'яті і коли буде “збір сміття”. Інструменти для аналізу батареї показують, яке навантаження припадає на пристрій.

Особливості Android Studio IDE[7]:

- розширений редактор макетів: WYSIWYG, можливість працювати з UI компонентами за допомогою функції Drag-and-Drop, функція попереднього перегляду макету на декілька конфігурацій екрану;
- зборка додатків, яка базується на Gradle;

- різні види зборок і генерація декількох .apk файлів;
- рефакторинг коду;
- статичний аналізатор коду, який дозволяє знаходити проблеми продуктивності, несумісності версій і таке інше;
- вбудований ProGuard і утиліта для підпису додатків;
- шаблони основних макетів і компонентів Android;
- підтримка розробки додатків Android Wear та Android TV;
- вбудована підтримка Google Cloud Platform, яка включає в себе інтеграцію з сервісами Google Cloud Messaging і App Engine;
- Android Studio підтримує Android N Preview SDK, а це означає, що розробники зможуть починати роботу по створенню додатків для нової програмної платформи;
- Android Studio може працювати з оновленим компілятором Jack.

До переваг Android Studio можна віднести:

- Середовище розробки підтримує роботу з декількома мовами програмування, до яких відносяться самі популярні C/C++, Java та Kotlin;
- редактор коду з яким зручно працювати;
- дозволяє розробляти додатки не тільки для смартфонів/планшетів, а і для портативних ПК, приставок для телевізорів Android TV, пристроїв Android Wear, нових мобільних пристроїв з незвичайним відношенням сторін екрану;
- тестування коректності роботи нових ігор, утиліт, їх продуктивності на тій чи іншій системі, виконується прямо в емуляторі;
- рефакторинг уже готового коду;
- достатньо велика бібліотека з готовими шаблонами і компонентами для розробки програмного забезпечення;
- великий набір засобів для тестування кожного елемента додатку або гри.

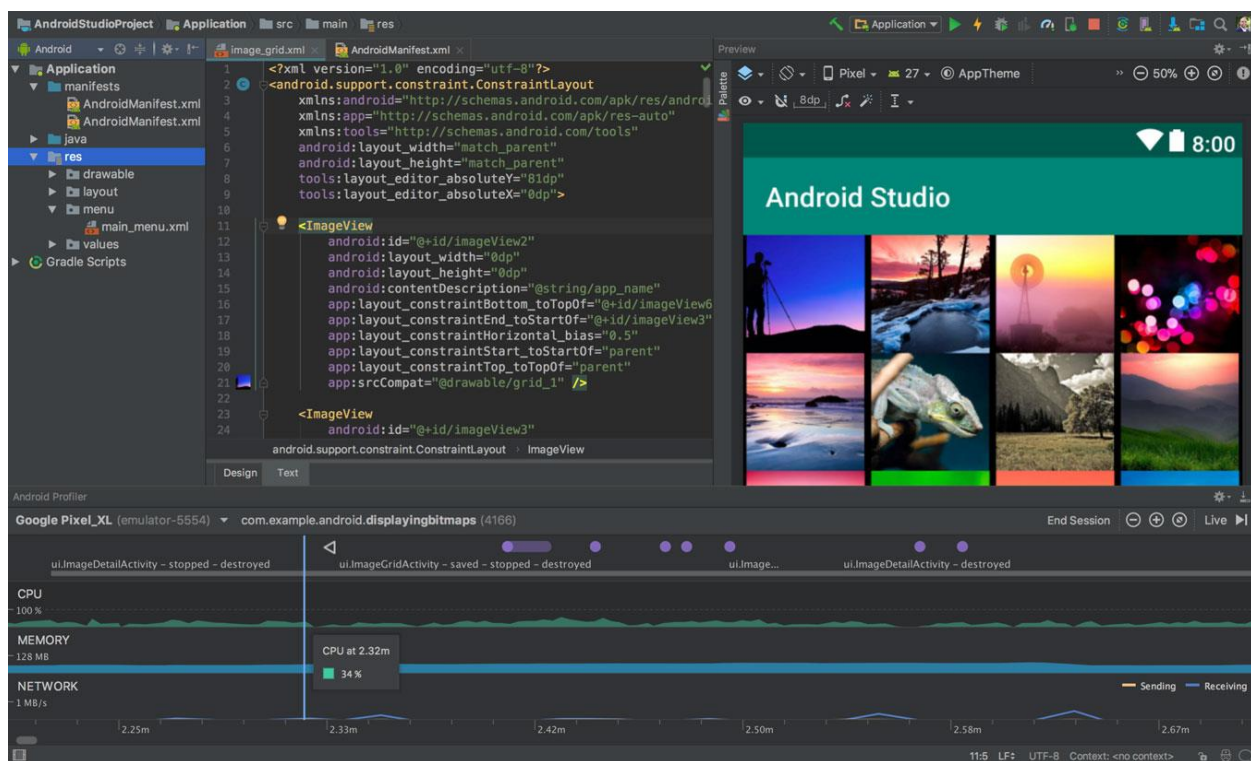


Рисунок 3.2 - Приклад роботи в Android Studio

Після вибору IDE для програмування під Android постало питання у виборі мови програмування. Вибір стояв між Java та Kotlin.

Kotlin – це нова статично типізована мова програмування з відкритим вихідним кодом. Він може ефективно запускатися на віртуальній машині Java (JVM). Kotlin розроблений JetBrains і офіційно підтримується Google.

На даний момент Kotlin використовується для створення Android додатків такими лідерами бізнесу як Pivotal, Atlassian, Pinterest, Evernote та Uber.

Для того щоб порівняти Java та Kotlin потрібно назвати переваги та недоліки кожної мови і порівняти ці дві мови програмування в цілому[5]:

- Виключення що перевіряються. Одна із основних різниць між Java та Kotlin полягає в тому що в останньому відсутня перевірка на виключення. Тому немає необхідності відловлювати і оголошувати будь-які виключення.

Зм.	Арк.	№ докум.	Підп.	Дата

- Стислість коду. В порівнянні класу Java з еквівалентним класом Kotlin демонструє лаконічність коду Kotlin. Для такої ж операції, яка виконується в класі Java, клас Kotlin потребує менше коду. Розширення Kotlin в Android дозволяє імпортувати посилання на View в файл Activity. Це дає можливість працювати з цим представленням так якби воно було частиною Activity. Це явно можна віднести до плюсів Kotlin.
- Функції розширення. Kotlin дозволяє розробникам розширювати клас новими функціями за допомогою функцій розширення. Такі функції не доступні в Java.
- Неявні розширювальні перетворення. В Kotlin немає підтримки неявних розширювальних перетворень для даних. Таким чином менші типи не можуть бути перетворені в більші типи. В той час як в Java підтримуються неявні перетворення, в Kotlin потрібно виконати саме явне перетворення.
- Не-private поля. Інкапсуляція необхідна в будь-якій програмі для досягнення бажаного рівня керування. Неприватні поля або public поля в Java корисні в сценаріях, де об'єкти повинні мінятися у відповідності з їх представленням. Це означає, що такі поля надають можливість представлення об'єкта, об'єктам що викликають. У Kotlin відсутні public поля.
- Статичні елементи. У Kotlin немає статичних елементів. Однак в мові програмування Java ключове слово static відображає те, що конкретний член, з яким використовується це ключове слово, належить самому типу, а не екземпляру цього типу. Це просто означає, що один і тільки один екземпляр цього статичного члена створюється і використовується всіма екземплярами класу.
- Підтримка конструкторів. Класи Kotlin, на відміну від класів Java, можуть мати один або кілька вторинних конструкторів, на додаток до

первинного конструктору. Це робиться шляхом включення цих вторинних конструкторів в оголошення класу.

- Потрійний оператор. На відміну від Kotlin, Java має умовний оператор. Умовний оператор в Java працює як базовий оператор if. Він складається з умови, яке оцінюється як істинне або помилкове. Крім того, умовний оператор в Java має два значення. Тільки одне з них повертається в залежності від того, чи є умова істинним або хибним. Синтаксис для умовного оператора Java:

(Стан)? (Значення 1): (значення 2)

- Бібліотеки обробки анотацій з Kotlin. Крім надання підтримки існуючим Java-фреймворки і бібліотекам, Kotlin також пропонує розширені Java-фреймворки, засновані на обробці анотацій. Однак застосування в Kotlin бібліотеки Java, яка використовує обробку анотацій, вимагає додавання її в проект Kotlin трохи іншим способом, ніж потрібно для бібліотеки Java, яка не використовує обробку анотацій.
- Нульова безпека. Однією з найпомітніших проблем для розробників, пов'язаних з Java, є NullPointerExceptions. Java дозволяє розробникам привласнити значення null будь-якої змінної. Однак, якщо вони намагаються використати посилання на об'єкт з значенням null, виникає виключення NullPointerException! На відміну від Java, в Kotlin всі типи за замовчуванням не є-nullable. Якщо розробники спробують привласнити або повернути значення null в коді Kotlin, під час компіляції відбудеться збій. Тим не менш, є спосіб обійти цей момент. Щоб привласнити значення null змінної в Kotlin, необхідно явно помітити цю змінну як nullable.

Очевидно, що деякі моменти краще реалізовані в Kotlin, в той час як для інших вигідно використовувати Java. Незалежно від всіх відмінностей між двома мовами програмування, вони повністю сумісні. І Java, і Kotlin

компілюються в байт-код. Але оскільки мова Kotlin все таки нова і ще місцями не зовсім допрацьована було прийнято рішення використовувати мову програмування Java.

3.3. Опис архітектури та принцип роботи

Проект складається з апаратного забезпечення та додатку для Android.

Керування і логіку апаратної частини було написано на мові програмування C++. Керування апаратною частиною здійснюється 3-ма основними функціями:

- Setup();
- Loop();
- Filter();

Розглянемо детальніше кожен функцію.

Функція Setup() – не приймає жодних аргументів і виконується лише один раз на початку роботи модуля для встановлення початкових даних у змінні, визначення режимів роботи виходів, запуску використовуваних бібліотек і так далі. Дана функція запускається тільки один раз після кожної подачі живлення.

Функція Loop() – викликається в нескінченному циклі відразу після одноразового виклику функції Setup(). Кожного разу вона виконує ті ж самі обчислення і реагує на них.

Функція Filter() – функція для фільтрування значень отриманих від акселерометра та гіроскопа.

Додаток для операційної системи Android має десяток класів.

Класи, які відповідають за з'єднання з апаратним забезпеченням по Bluetooth протоколу:

- клас `BluetoothConnectionService`. Цей клас відповідає за проведення з'єднання і роз'єднання між телефоном користувача та пристроєм. До складу даного класу входять ще декілька класів. Перший клас `AcceptThread`, який реалізовує потік підтвердження з'єднання в захищеному режимі. При запуску цього потоку відбувається включення Bluetooth на телефоні користувача і відбувається створення сокету для забезпечення захищеної передачі даних між телефоном і пристроєм. Наступний підклас у класі `BluetoothConnectionService` це клас `ConnectThread`. Цей клас як і попередній реалізує потік завдяки якому відбувається підключення до пристрою. Відбувається з'єднання по сокету який був визначений у `AcceptThread`. І останній підклас це `ConnectedThread`. Даний клас який реалізує потік на початку і протягом всієї своєї роботи відхиляє всі сторонні запити на підключення до телефону. І в цьому ж потоці виконується обмін даними між пристроєм та телефоном.
- клас `DeviceListAdapter`. Даний клас виконує пошук всіх Bluetooth модулів поблизу і виділяє всі їхні дані. Ці дані містять ім'я та адресу модулю.

`MainActivity` головний клас проекту у якому є меню для взаємодії з користувачем. Дане меню містить шість пунктів:

- Перший із них реалізований у класі `FragmentDashboard`. Даний клас відповідає за відображення поточних даних про швидкість і найкращі показники на різних дистанціях. Дані, які надходять на телефон в першу чергу відправляються в цей клас для швидкого відображення.

- Другий пункт меню реалізований у класі `FragmentResults`. У цьому класі відбувається відображення результатів всіх можливих проміжків для заміру даних (0 км/год – 100 км/год, 50 км/год – 150 км/год, 80 км/год – 180км/год, 100 км/год – 200 км/год, ¼ милі, ½ милі, 1 кілометр і 1 миля). Також

в цих полях окрім ідентифікатора проміжку міститься поле в якому відображена кількість замірів в даному проміжку. `FragmentResults` при виклику метода `onCreate()` отримує дані від `MainActivity` про всі доступні проміжки і відповідним чином відображує їх.

- Наступний клас `FragmentUnits`, який зв'язаний з третім пунктом меню. Даний клас як і попередній також реалізує відображення всіх можливих проміжків. Але з єдиною відмінністю, що в цьому фрагменті при виборі якогось з проміжків відбувається відключення замірів на даному проміжку також це одразу фіксується у головному `Activity`, який у свою чергу передає дані у `Fragment Dashboard` та `Fragment Results`. Таким чином відбувається не пряма взаємодія між фрагментами меню.

- Клас `FragmentMyCar` завдяки якому відбувається взаємодія між користувачем та додатком. Даний клас призначений для вводу даних користувачем про свій автомобіль і подальшого їх збереження. Введені дані зберігаються в локальний файл і при наступному запуску програми відразу відбувається зчитування даних з відповідного файлу і потім відображаються у відповідних полях.

- Клас `FragmentFavourite`, який напряду пов'язаний з п'ятим пунктом меню. Даний клас реалізовує відображення збережених даних про відібрані значення заїздів. Користувач в ході користування додатком може вподобати якісь дані і вибрати їх. Ці дані зберезуться в файл `results.txt` з позначкою `like`. При запуску п'ятого пункту меню `Fragment Favourite` виконує пошук у вищезгаданому файлі і залишає лише дані із поміткою `like` і потім відображує їх у вигляді списку.

- Клас `FragmentSettings` відповідає за налаштування додатку.

- Клас `IntervalResultsActivity` відповідає за відгук на вибір користувачем одного з проміжків у `FragmentResults`. При виборі одного з проміжків даних

методом putExtra() відправляються дані про вибраний сегмент і виконується запуск IntervalResultsActivity. При відкритті даного Activity методом getExtra() витягуються дані, які надійшли з попереднього Activity в яких зазначено проміжок, за яким потрібно провести пошук у файлі і відобразити дані які співпадають з заданим проміжком.

Оскільки всі вищезазначені класи являються класами додатку для Android це означає, що у них є життєвий цикл Activity та спільні стандартні методи, які забезпечують і описують цей життєвий цикл[10][15].

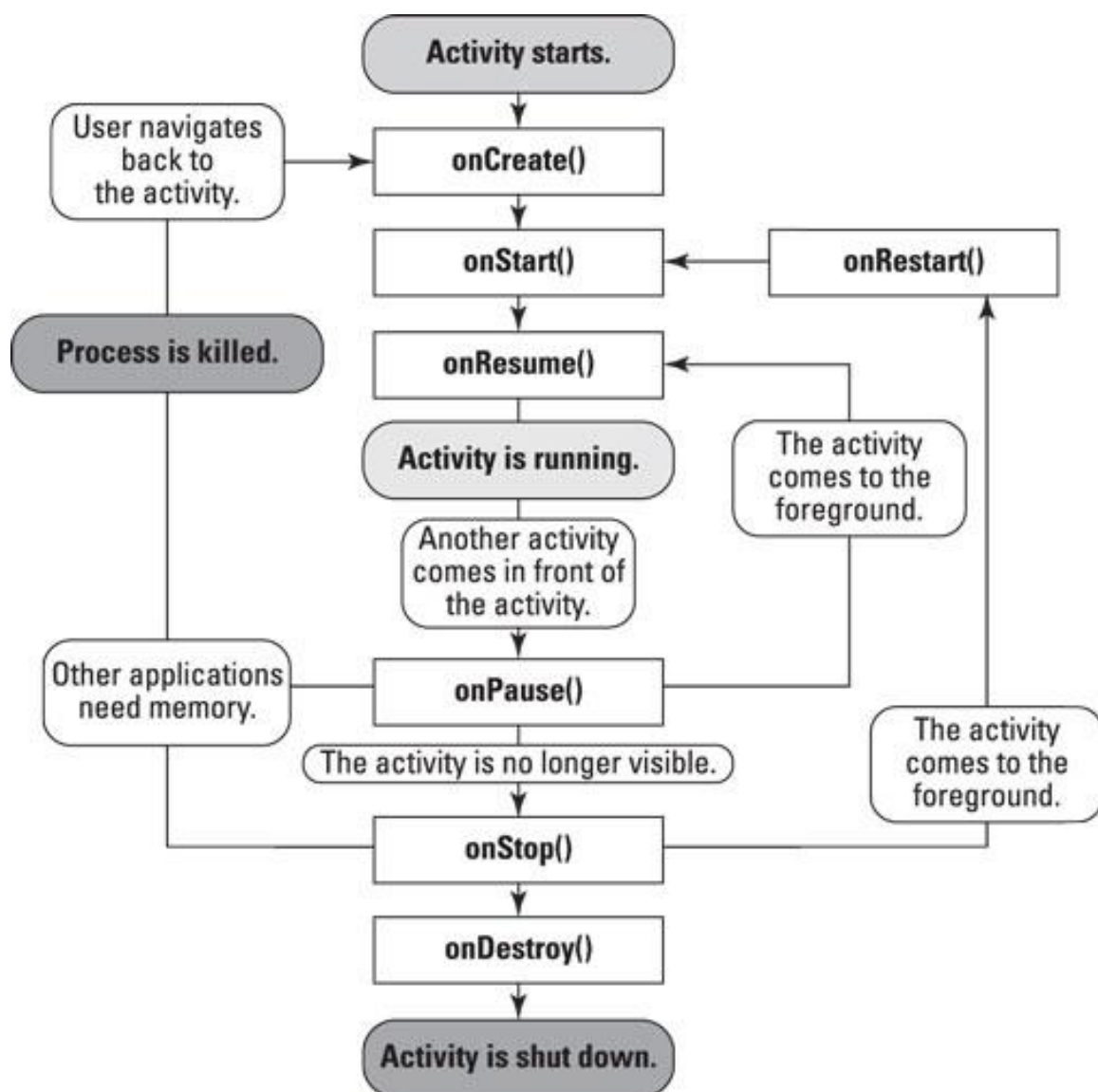


Рисунок 3.3 - Життєвий цикл Activity

При запуску Activity викликається метод onCreate(). Якщо зберігається стан додатку в пакеті, його можна повернути назад в onCreate(), якщо необхідно активувати подію (наприклад, зміна орієнтації). Якщо орієнтація змінюється (тобто пристрій перевертається із ландшафтного режиму на портретний і навпаки), активність відтворюється і метод onCreate() викликається знову, щоб не втратити цю попередню інформацію.

Метод onRestart(). Даний метод викликається після зупинки операції перед її повторним запуском. За ним слідує метод onStart().

Метод onStart(). Викликається безпосередньо перед тим як операція стає видимою для користувача. За нею слідує метод onResume(), якщо операція переходить на передній план, або метод onStop(), якщо вона стає невидимою.

Метод onResume(). Викликається безпосередньо перед тим, як операція починає взаємодіяти з користувачем. На цьому етапі операція знаходиться на самому верху стеку операцій і в неї не надходять дані, які вводить користувач. За ним завжди слідує метод onPause().

Метод onPause(). Система викликає даний метод в якості первинної ознаки виходу користувача із операції(проте це не завжди означає, що операція буде знищена). Зазвичай саме тут необхідно застосовувати будь-які зміни, які повинні бути збережені крім поточного сеансу роботи користувача (оскільки користувач може не повернутися назад). За ним слідує або метод onResume(), якщо операція повертається на передній план, або метод onStop(), якщо операція стає скритою для користувача.

Метод onStop(). Даний метод викликається у випадку коли операція більше не відображається для користувача. Це може відбутися по причині того, що операція знищена, або через те що оновилася поверх неї інша операція (існуюча або нова). За ним слідує або метод onRestart(), якщо

операція поновлює взаємодію з користувачем, або метод onDestroy(), якщо операція переходить в фоновий режим.

Метод onDestroy(). Викликається перед тим, як операція буде знищена. Це фінальний виклик, який отримує операція. Його можна викликати або через завершення операції (виклик методу finish ()), або внаслідок тимчасового знищення системою цього примірника операції з метою звільнити місце. Щоб розрізнити ці два сценарії, використовується метод isFinishing ().

3.4. Опис інтерфейсу

При запуску додатку автоматично відкривається вкладка з даними про поточну швидкість та найкращі заміри результатів.

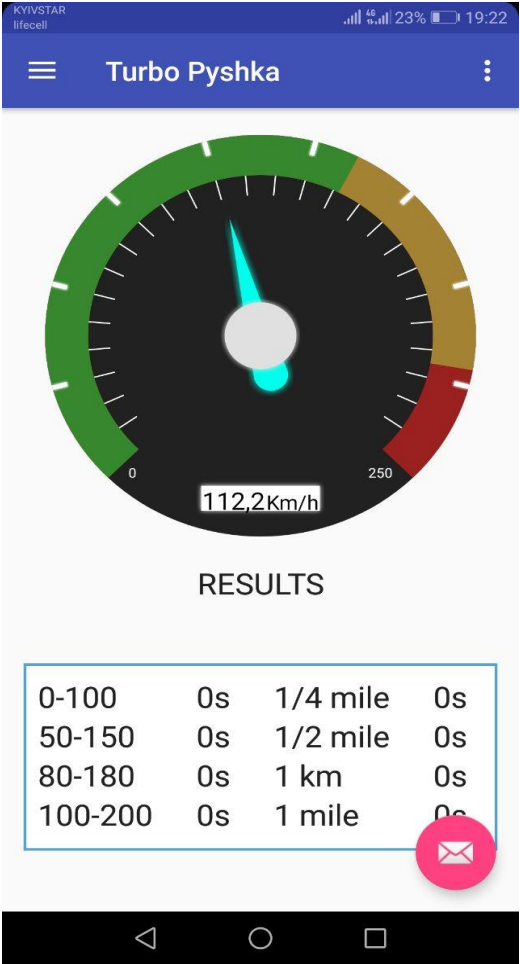


Рисунок 3.4 - Початковий екран додатку

Як видно з рисунку 3.4 при запуску додатку користувача зустрічає спідометр, який показує швидкість на даний момент. Знизу від спідометра в таблиці знаходяться найкращі дані з усіх можливих проміжків. У правому куті можна вибрати меню за допомогою якого можна приєднатися, або від'єднатися від пристрою. Натиснувши на верхній лівий кут або зсувом вправо користувач може викликати меню, яке показане на рисунку 3.5.

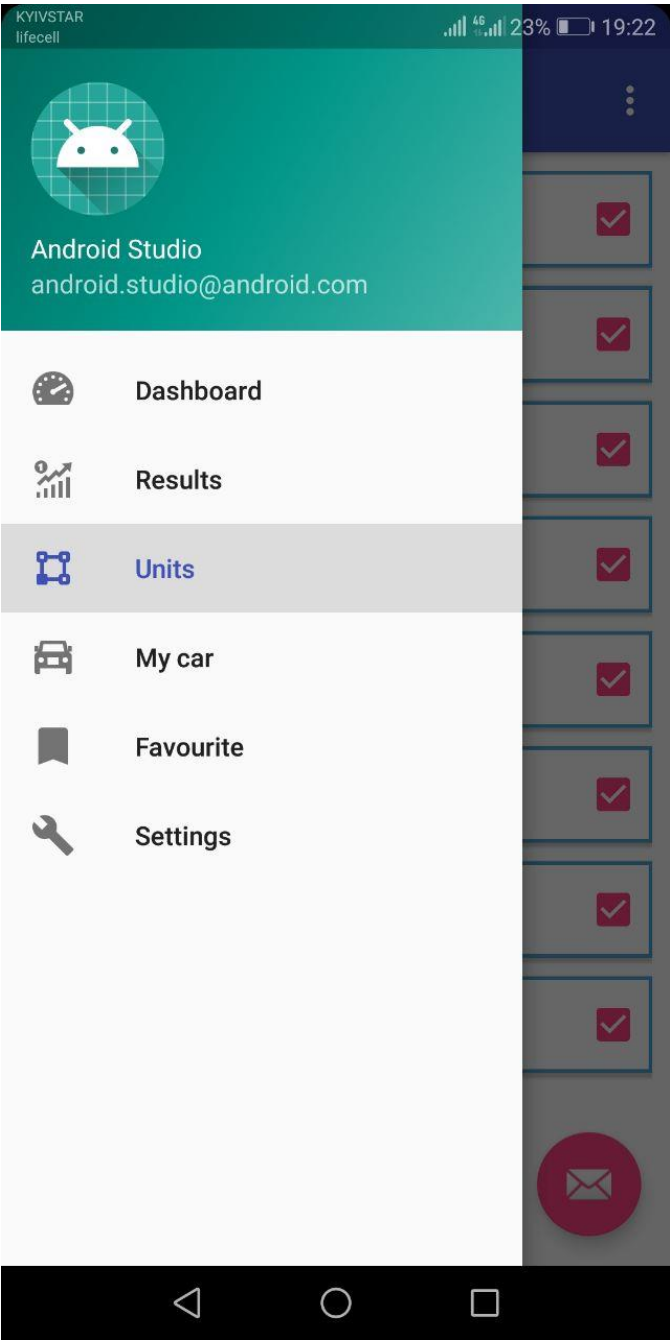


Рисунок 3.5 - Допоміжне меню

Як видно з рисунку меню має 6 пунктів. Пункт Results (рис. 3.6) призначений для відображення всіх проміжків на яких виконуються заміри та кількість замірів, яка вже була проведена.

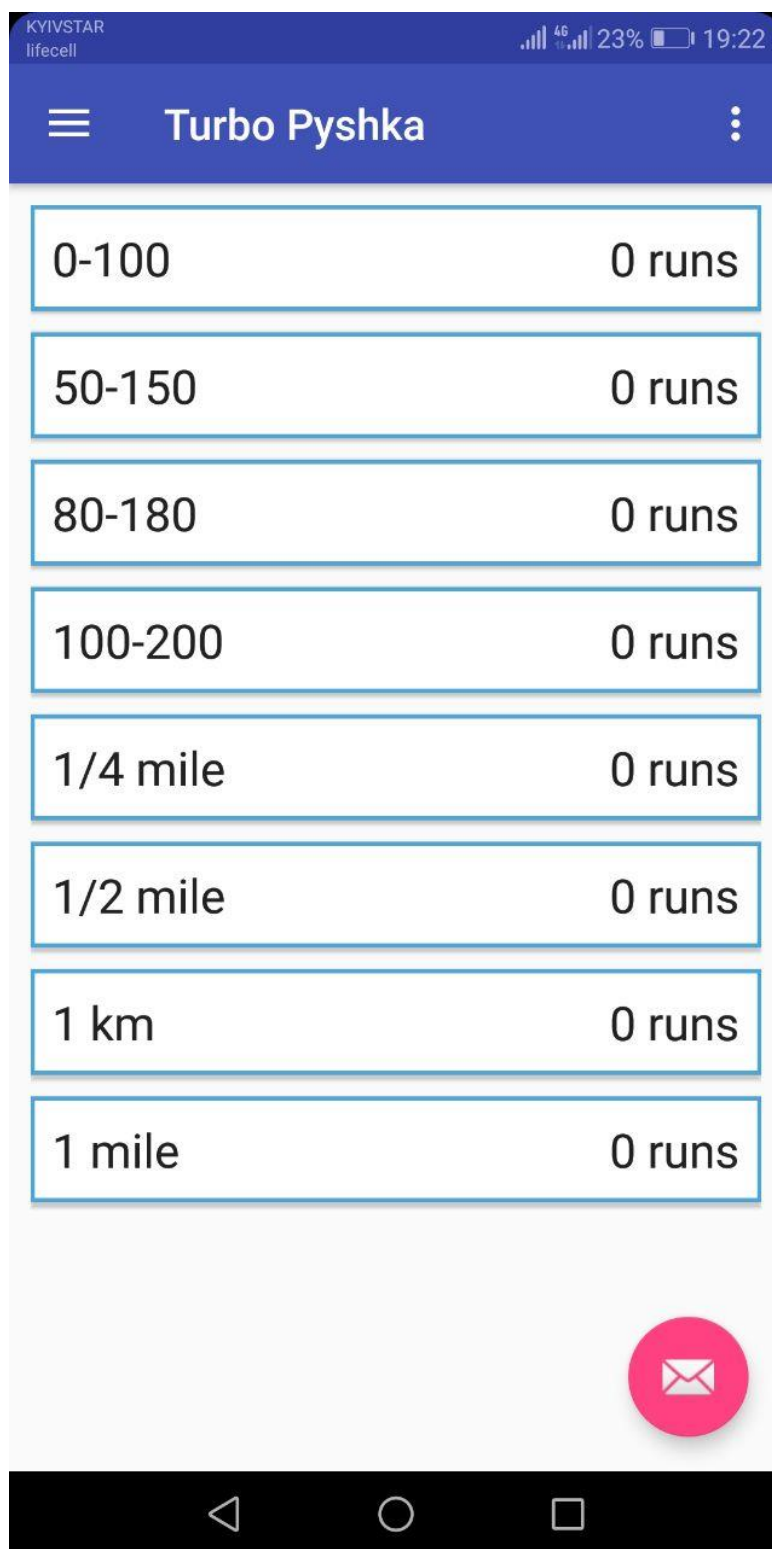


Рисунок 3.6 - Зміст пункту меню Results

Обравши і натиснувши на будь-який проміжок користувач має змогу побачити більш детальні дані з обраного проміжку (рис. 3.7) . Буде показана детальна інформація, така як: проміжок який замірявся, дата коли відбувався замір даних, час та результат заміру, який вимірюється у секундах.

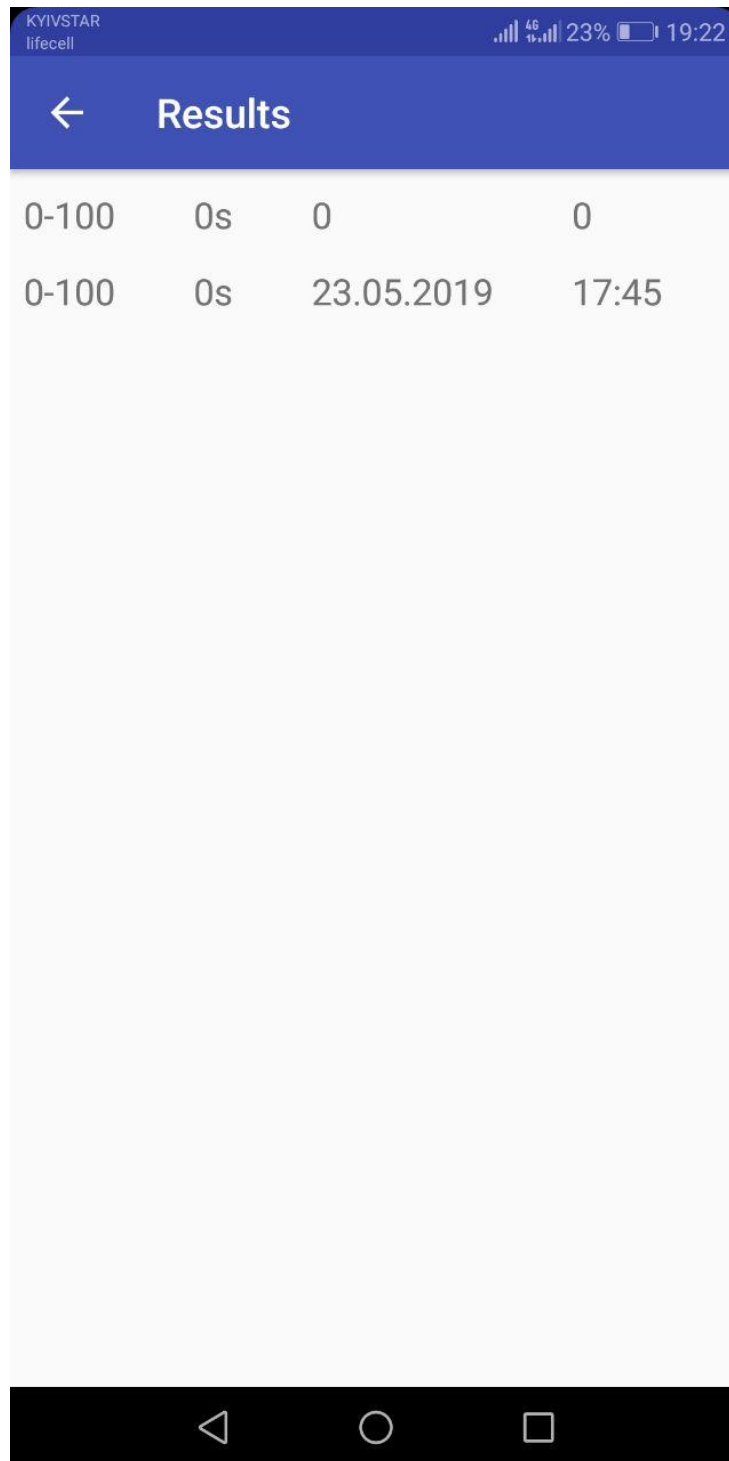


Рисунок 3.7 - Детальне відображення результатів на проміжку

Також звичайно користувач може вибрати, які з даних будуть замірятися при наступному заїзді (рис. 3.8) . Для цього є третій пункт меню Units. В якому користувач може натиском на “галочки” відключити або підключити потрібні йому заміри. Відповідно якщо прибрати якийсь проміжок в Units то він пропаде і з Results і не буде відбуватися замір при наступному замірі показників.

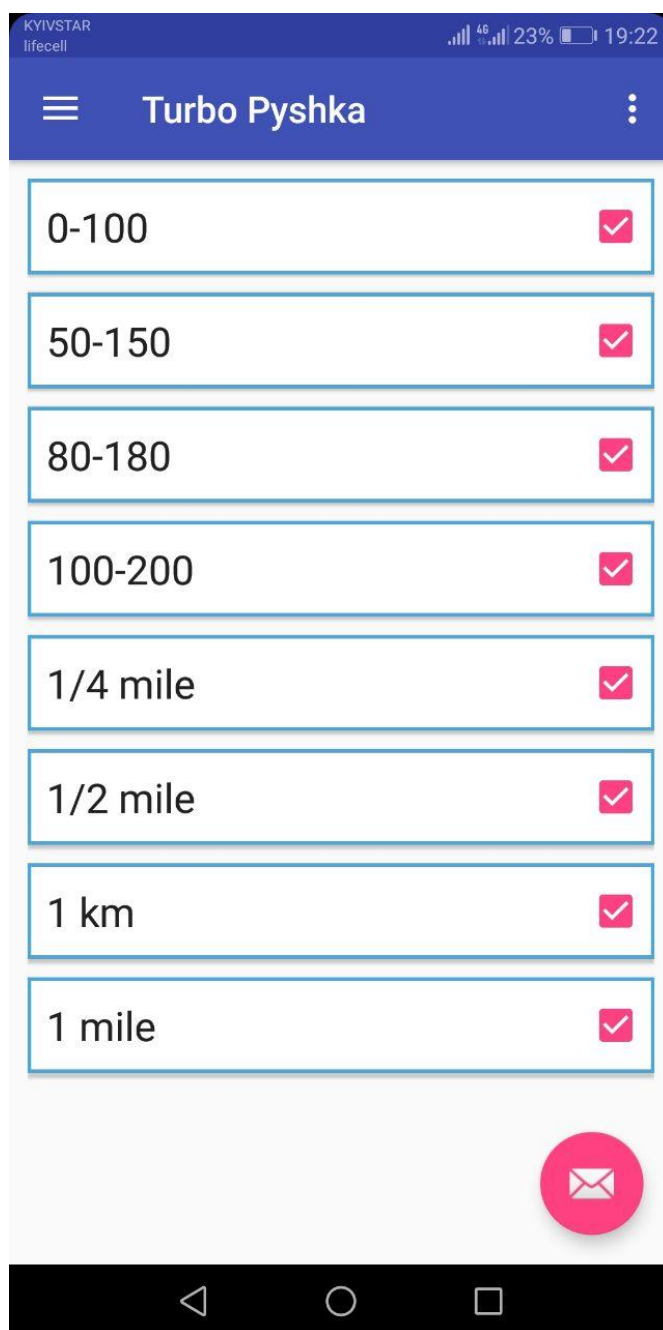


Рисунок 3.8 – Фрагмент меню Units

4. ВИСНОВОКИ

В ході виконання даного дипломного проекту було досліджено сферу пристроїв для вимірювання динаміки автомобіля. В ході дослідження було проаналізовано ряд найкращих пристроїв, які представлені на даному ринку та створено власний пристрій на базі GPS для вимірювання динаміки автомобіля.

У першому розділі було проаналізовано загальний опис проблеми, готові рішення, які існують. Дослідження показали, що проблема з пристроями для вимірювання автомобілів існує та досить актуальна в наш час. Були розглянуті основні переваги та недоліки вже існуючих пристроїв, які якимось чином вирішували дану проблему і не в повній мірі розв'язували її.

Другий розділ дипломного проекту присвячено особливостям розробки DRT, а саме наведені три різні конфігурації, які були дослідженні в ході розробки пристрою, проаналізовані основні переваги та недоліки. В даному розділі вибрано остаточну схему моделі та описано основні особливості та переваги приладу.

В останньому розділі було описано весь процес розробки апаратного та програмного забезпечення. Крім того, описано принцип роботи розробленої схеми, взаємодію основних модулів та інтерфейс користувача.

Поставлена задача була виконана успішно та має право на існування та своє подальше удосконалення. Даний проект вже зараз може бути випущений в великій кількості і може достойно конкурувати з вже представленими на ринку пристроями. Варто зазначити, що архітектура даного рішення не можна кастомізувати, оскільки архітектура розробленого пристрою є остаточною.

5. СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Белов Олександр Створюємо пристрої на мікроконтролерах., 2007
2. Блох Джошуа Ефективне програмування., 2014
3. Брайан Харді, Білл Філліпс, Кріс Стюарт, Крістін Марсікано. Програмування під Android. 2-е видання., 2016
4. Віктор Петін Проект з використанням контролера Arduino., 2015
5. Герберт Шілдрт Java 8. Повне керівництво 9-е видання., 2015
6. Дарвін Ян Android. Збірник рецептів. Задачі і рішення для розробників додатків., 2016
7. Дейтел Пол, Дейтел Еббі, і Харві Дейтел Android для розробників., 2014
8. Джордж Хайнеман, Гарі Полліс, Стенлі Селков Алгоритми. Довідник з прикладами на C, C ++, Java і Python., 2017
9. Джошуа Керіевскі Рефакторинг з використанням шаблонів., 2008
10. Лі Атчісон Масштабування додатків. Вирощування складних систем., 2016
11. Метт Вайсфельд Об'єктно-орієнтоване мислення., 2014
12. Саймон Монк Програмуємо Arduino. Основи роботи зі скетчами., 2016
13. Стівен Прата Мова програмування C++. Лекції і вправи., 2015
14. Янта Катупітія, Кім Бентлі Управління електронними пристроями на C++ ., 2016
15. Ed Burnette Hello, Android: Introducing Google's Mobile Development Platform., 2009